Leandro Nunes de Castro
Fernando José Von Zuben
Helder Knidel (Eds.)

LNCS 4628

Artificial Immune Systems

6th International Conference, ICARIS 2007
Santos, Brazil, August 2007
Proceedings

Springer

Lecture Notes in Computer Science 4628

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

Leandro Nunes de Castro
Fernando José Von Zuben Helder Knidel (Eds.)

Artificial Immune Systems

6th International Conference, ICARIS 2007
Santos, Brazil, August 26-29, 2007
Proceedings

Springer

Volume Editors

Leandro Nunes de Castro
Catholic University of Santos, UniSantos
R. Dr. Carvalho de Mendonça, 144, Vila Mathias, 11070-906, Santos/SP, Brazil
E-mail: lnunes@unisantos.edu.br

Fernando José Von Zuben
DCA/FEEC/Unicamp
Caixa Postal 6101, Campinas/SP, 13083-970, Brazil
E-mail: vonzuben@dca.fee.unicamp.br

Helder Knidel
NatComp - From Nature to Business
R. do Comércio, 44, Sala 3, Santos/SP, 11010-140, Brazil
E-mail: helder.knidel@natcomp.com.br

Library of Congress Control Number: 2007931604

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 3-540-73921-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-73921-0 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12099177 06/3180 5 4 3 2 1 0

Preface

The field of artificial immune systems (AIS) is one of the most recent natural computing approaches to emerge from engineering, computer science and theoretical immunology. The immune system is an adaptive system that employs many parallel and complementary mechanisms to maintain homeostasis and defend the organism against pathological agents. It is a distributed system, capable of constructing and maintaining a dynamical and structural identity, learning to identify previously unseen invaders and remembering what it has learnt. Numerous immune algorithms now exist, based on processes identified within the vertebrate immune system. These computational techniques have many potential applications, such as in distributed and adaptive control, machine learning, pattern recognition, fault and anomaly detection, computer security, optimization, and distributed system design.

The International Conference on Artificial Immune Systems (ICARIS) started in 2002 with the goal of bringing together a number of researchers investigating forms of using ideas from the immune system to do engineering and computing and to solve complex problems. Some theoretically oriented researchers also joined this effort with ambitious goals such as modeling the immune system. There is a continued effort to strengthen the interaction among distinct research areas, aiming at supporting the multidisciplinary outline of the field. Table 1 indicates the number of submissions versus the number of published papers for each of the six ICARIS conferences up to now. From 2004 to 2007 the number of submissions and accepted papers has varied little with a slight increase in 2005, although one would probably expect these numbers to have increased more over time, due to the existence of mature textbooks and survey papers in the literature. Despite that, the submissions this year came from 24 countries (Lithuania, Switzerland, Luxemburg, Chile, Taiwan, Japan, Malaysia, Morocco, Iran, Portugal, Belgium, Algeria, Turkey, Poland, India, Pakistan, Colombia, USA, Hong Kong, Germany, Republic of Korea, P. R. China, UK and Brazil), and the range of innovative and well-succeeded applications of immune-inspired algorithms is increasing significantly. As we are with the field almost from its inception, we noticed that ICARIS conferences are playing a great role in bringing newcomers to the field. It is a challenge for us as a community to stimulate these newcomers and encourage others, so that the field may face sustainable growth and progress.

Concerning the event organization, for us it was a great pleasure to host ICARIS in Santos/SP, Brazil. This is a particularly interesting city in Brazil, for it contains the largest port in Latin America, it is surrounded by paradisiacal beaches and dense Atlantic forests, and it is the house of one of the most traditional Brazilian soccer teams: Santos Futebol Clube, the soccer team where Pele, the most famous soccer player around the world, developed his splendid career.

Table 1. Number of submissions versus number of accepted papers for each ICARIS conference

Year	Submissions	Acceptance (Rate%)
2002	—	26 (—%)
2003	41	26 (63%)
2004	58	34 (59%)
2005	68	37 (54%)
2006	60	35 (58%)
2007	58	35 (60%)

ICARIS 2007 provided a number of activities for its attendees, from lectures, to tutorials, software demonstrations, panel discussions, and paper presentations. We had the pleasure of bringing Rob de Boer (University of Utrecht, Netherlands), Jorge Carneiro (Instituto Gulbenkian de Ciências, Portugal), Hugues Bersini (IRIDIA, Brussels), and Uwe Aickelin (University of Nottingham, UK), for the event.

The organization of ICARIS 2007 would not have been possible without the support of a number of committed institutions and people. We are particularly indebted to our home institutions and company, UniSantos, Unicamp and NatComp, respectively, and to all the collaborators and sponsors that helped to make ICARIS 2007 a success.

August 2007

Leandro Nunes de Castro
Fernando Von Zuben
Helder Knidel

Organization

ICARIS 2007 was organized by the University of Santos (UNISANTOS), State University of Campinas (UNICAMP) and NatComp - From Nature to Business.

Executive Committee

Conference Chairs	Leandro Nunes de Castro (UniSantos, Brazil)
	Fernando J. Von Zuben (Unicamp, Brazil)
Conference Secretary	Helder Knidel (NatComp, Brazil)
International Advisory Board	Jonathan Timmis (University of York, UK)
	Emma Hart (Napier University, UK)
	Hugues Bersini (IRIDIA, ULB)
	Steve Cayzer (Hewlett-Packard, UK)
Publicity Chairs	Carlos A. Coello Coello (CINVESTAV, Mexico)
	Dipankar Dasgupta (University of Memphis, USA)
	Ernesto Costa (University de Coimbra, Portugal)
	Siti Zaiton Mohd Hashim (Universiti Teknologi Malaysia, Malaysia)
	Yoshitero Ishida (Toyohashi University of Technology, Japan)

Referees

A. Freitas
A. Tarakanov
A. Watkins
A. Tyrrell
C.C. Coello
C. Johnson
D. Flower
D. Dasgupta
D. Lee
E. Hart
E. Costa
F. Gonzalez
F. Esponda
F.J. Von Zuben
F. Castiglione
G. Nicosia
H.Y.K. Lau
H. Bersini
J. Timmis
J.A. Costa
J. Carneiro
J. Kim
L.N. de Castro
M. Neal
M.R.B.S. Delgado
P. Arena
P. Vargas
P. Bentley
P. Ross
S. Garrett
S.Z.M. Hashim
S.T. Wierzchon
S. Forrest
S. Cayzer
S. Stepney
T. Stibor
U. Aickelin
V. Cutello
W. Caminhas
W. Luo
Y. Ishida

Sponsoring and Support Institutions

Capes
CNPq
Energisa S/A
Esférica Tecnologia
Fapesp
Hewlett-Packard
NatComp
Petrobrás
SAE Institute
SBA
SBC
Unicamp
UniSantos

Table of Contents

Search and Optimization

Classification and Clustering

Anomaly Detection and Negative Selection

Robotics, Control and Electronics

Modeling Papers

Conceptual Papers

Technical Papers and General Applications

A Gradient-Based Artificial Immune System Applied to Optimal Power Flow Problems

Leonardo de Mello Honório, Armando M. Leite da Silva, and Daniele A. Barbosa

Federal University of Itajubá, Minas Gerais, Brazil
{demello,armando,danieleb}@unifei.edu.br

Abstract. Mathematically, an optimal power flow (OPF) is in general a non-linear, non-convex and large-scale problem with both continuous and discrete control variables. This paper approaches the OPF problem using a modified Artificial Immune System (AIS). The AIS optimization methodology uses, among others, two major immunological principles: hypermutation, which is responsible for local search, and receptor edition to explore different areas in the solution space. The proposed method enhances the original AIS by combining it with a gradient vector. This concept is used to provide valuable information during the hypermutation process, decreasing the number of generations and clones, and, consequently, speeding up the convergence process while reducing the computational time. Two applications illustrate the performance of the proposed method.

Keywords: Artificial immune system, gradient-based algorithms, optimal power flow, transmission loss reduction.

1 Introduction

One of the most important areas in electric power systems is the study of optimal power flow problems. There are several relevant applications of this tool for planning the power network expansion, operation, maintenance and, most recently, for solving different problems in the emerging electricity markets [1]-[11]. Moreover, many applications in power system reliability use OPF as the major tool of their analyses [12], [13]. In general, an OPF is a non-linear, non-convex and large-scale problem involving several sets of continuous and discrete variables. This diversity makes the OPF problem to be divided, according to solution space, convexity, and types of control variables, into several broad categories such as linear, non-linear, combinatorial, dynamic, probabilistic, and others. In order to deal with these categories, different optimization concepts or methods have been employed [1]-[11], [14]-[19]: Simplex, Interior-Point, Conjugate Gradient, Hill Climbing, Tabu Search, Genetic Algorithm, Ant Colony, Particle Swarm Optimization, Artificial Immune System, etc.

These methodologies can further be divided into two major groups: numerical and intelligent-based. Regarding numerical-based methodologies, reference [5] shows a comparison among three interior-point-based methods, primal-dual (PD),

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 1–12, 2007.

predictor-corrector (PC) and multiple-centrality-correction (MCC). The results show good performance for all methods especially the MCC, although it needs accurate parametric adjustments to improve the convergence performance. Reference [9] uses the PC and PD to test different control actions and objective functions in order to enhance systems loadability.

Although these conventional methods have presented good results, some drawbacks have appeared in actual power system applications. Reference [7] shows that handling discrete variables as continuous ones, until they are close to the optimal solution, and then rounding them off to the nearest corresponding values, may provide significant higher cost results than the one present by the actual optimal solution. It also suggests that techniques capable of working properly with mixed-integer programming models may suffer from scalability and, therefore, are unsuitable for large-scale power systems. On the other hand, intelligent-based methodologies are interesting alternatives for some of the discussed OPF problems, as it can be seen recently in the power system literature: e.g., Particle Swarm [4], [10], Genetic-Algorithm [11], Greedy search [6], and many others. Several advantages can be linked to those methods; the software complexity is simple; they are able to mix integer and non-integer variables, and also present very appealing computational performance. However, the problem with many of these methodologies is the difficulty in establishing the Karush-Kuhn-Tucker conditions [3], [5], at the end of the optimization process. One method able to deal with this problem is the Artificial Immune System. Although very few AIS applications in power flow optimization can be found in the literature [14]-[16], results in other engineering fields [20]-[23] are very promising, and must encourage power engineers to further explore these techniques.

The AIS method is based on the biological principle of bodies' immune systems [20], [21]. An immunological system has major characteristics that can be used in learning and optimization [22]. For optimization problems, three topics are especially interesting: proliferation, mutation, and selection. While proliferation is the capability of generating new individuals making the optimization process dynamic, mutation is the ability of searching through the solution space for sub-optimum points. The selection is responsible for eliminating low-affinity cells. These three features make AIS a powerful optimization tool, enabling the search for several local-optima.

There are several variants among AIS methodologies, available in the literature, used to implement optimization algorithms. Reference [22] shows a very interesting approach by embedding a useful property of evolutionary algorithms, *niching,* which drives individuals to the most promising points in the solution space. Although this algorithm has exhibited very good results, the number of individuals used in the simulation processes is very high bearing in mind OPF problems. Since the purpose of this paper is to demonstrate the effectiveness of AIS concepts in OPF problems, some modifications in the referred algorithm are being proposed by adding more relevant information to individuals. In an electric power system, an individual can be related to an operating condition, which is characterized by the corresponding power flow equations that describes its electrical behavior [24], [25]. Therefore, when a modification in any control variable is performed, it is possible to predict the associated operating point by analyzing the behavior of the power flow equations. Mathematically, by using the information given by the tangent/Jacobian vector (i.e. gradient) associated with these equations, it is possible to lead the mutation process, which in the original AIS algorithm was made through a completely random approach, to generate better individuals, making the optimization much faster and more reliable. In order to demonstrate the effectiveness of the proposed Gradient-Based AIS, an optimal power flow aiming to minimize network transmission losses is implemented.

This paper is organized as follows. Section 2 describes the optimal power flow problem and formulation. Section 3 shows the implementation of the proposed Gradient-Based AIS and some tests using as benchmark reference [22]. Section 4 illustrates and discusses the OPF results, obtained with an IEEE standard test system.

2 The Optimal Power Flow Problem

It is not the purpose of this work to present a full explanation on electric power system static and dynamic behaviors; for that references [24] and [25] are much more appropriate. Instead, the main concepts and the corresponding mathematical models are summarized as follows.

An electric power system is composed of several electrical equipment such as generators, transmission lines, transformers, shunt capacitors, etc. Its main goal is to generate, transmit, and distribute electric power to costumers, through a high quality service at minimum cost. There are several constrains linked to this aim. The most important one is that it is not possible to store electrical power and, therefore, the amount of generated energy at any given time must be the same as the amount consumed, duly discounted the transmission losses. In order words, the power flow balance must be null. To fulfill these conditions, human operators must handle hundreds, sometimes thousands, of variables in order to control the system driving energy flow from sources to consumers. Moreover, during this process, the system must remain physically stable, from both static and dynamic points of view, technically reliable, and economically interesting for all market agents involved in this process.

The previously described balance conditions can be mathematically stated as follows:

$$\begin{aligned} &y = M(x_{cl}) \\ &g_I(V,\theta,y,x_{cb}) - g_G + g_L = 0 \end{aligned} \tag{1}$$

where: y is the system admittance matrix; x_{cl} is the control variables related to the models of the transmission elements (e.g. lines and transformers); x_{cb} the control variables related to the models of the buses (i.e., power stations); g_I, g_G and g_L are the vectors of power injections, power generations and power loads at each bus, respectively. If any change happens in the control variables, the system is driven into a different operating point, changing values of several measures such as voltage magnitudes V and angles θ (named as state variables), and also active/reactive flows, etc., spread along the power network. This leads to a condition where finding the best operation point under a desirable scenario is challenging. This problem is named as optimal power flow or simply OPF.

As previously stated, an OPF is a non-linear, non-convex and large-scale problem, mathematically described as follows:

$$\begin{array}{ll} \textit{Minimize} & f(x_s, x_c) \\ \textit{Subject to} & g(x_s, x_c) = 0 \\ & \underline{H} \le h(x_s, x_c) \le \underline{H} \\ & \overline{x}_s, \overline{x}_c \le x_s, x_c \le \underline{x}_s, \underline{x}_c \end{array} \tag{2}$$

where: x_s is a *ns×1* vector of state variables (voltage magnitudes and angles at load buses, etc.); x_c is a *nc×1* vector of control variables (voltage magnitudes at generation buses, power generation, shunt capacitor allocation, transformer taps, load in buses, etc.); *f* is an scalar function representing a power system quantity to be optimized (e.g. economic dispatch, transmission loss reduction, transmission equipment overflow, loadability, load shedding); *g* is the active and reactive power balance equations; and *h* is a *m×1* vector of constraints associated with the limits of some network power values such as transmission lines flows, reactive generation, etc.

As previously stated, the literature shows several approaches to solve (2), and these can be divided into two major groups; numerical- and intelligent-based methods. The most commonly used numerical-based methods change the original problem into a *Lagragean* function and try to minimize it. In a different way, most intelligent-based methods use *brute force* computation associated with metaheuristics in order to narrow the search. This is related to population-based methods that generate several individuals, being each one a feasible or unfeasible system solution. The optimization process continues through interactions among those individuals, where the form of those interactions depends on the chosen methodology.

For an OPF problem, a feasible solution is a set of control actions that fulfill all constrains shown in (2). However, to find a feasible solution demands the execution of a numerical iterative method in order to solve (1), usually the Newton-Raphson [24]-[25]. Thus, the solution design of a population-based method is shown in Figure 1, where: *(a) New Population* – is the set of control actions that defines the individuals; *(b) Newton Raphson* – uses this solution process to, given a set of controls, define the system operating point; *(c) Validate Limits* – checks if the solution provided by each individual obeys the operation constraints and invalidates the out-of-range ones; *(d) Evaluate* - validates individuals according to (2); *(e) End* – if it is the end of the optimization process, pick up the best individual, otherwise, evolve the population and return to step *(b)*.

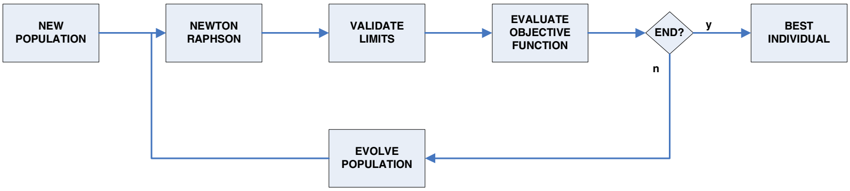


Fig. 1. Diagram Solution of a Population-Based Method

3 Gradient-Based Artificial 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 exposition 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 to attack the Ag's, or Memory cells, which store characteristics of the successful Ab's, providing a faster immunological response, when further expositions to the same Ag occurs. Figure 2 illustrates this process.

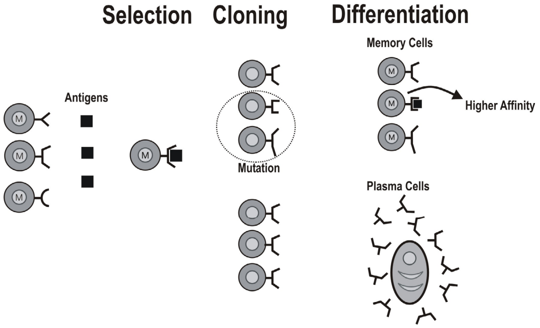


Fig. 2. Natural Immune System Diagram

The Artificial Immune System (AIS) intends to capture some of the principles previously described within a computational framework. The main purpose is to use the successful NIS process in optimization and learning. As 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 simultaneously local and global searches. These tools are based on two concepts: hypermutation and receptor edition. While hypermutation is the ability to execute small steps toward 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 of the most interesting is the CLONALG algorithm presented in [22]. The main statement of CLONALG is that progressive adaptive changes can be achieved by cumulative blind variation, based only upon an affinity increase caused by random mutation and individual selection. It also states that, through these principles, it is possible to generate high-quality solutions to complex problems.

Although the CLONALG algorithm has shown good results to different types of problems, the solution is entirely provided by computational *brute force*; hundreds of individuals are partially generated by random. The straightforward application of these concepts to OPF problems, where each individual represents a possible operating condition assessed by solving the power flow equations, would demand a huge amount of computational effort. Thus, instead of using only *brute force*, numerical information of the system, obtained as a by-product of the power flow solution assessments, will lead to a significant reduction in the number of clones and, consequently, in computing effort. Simple numerical information to be used is the first order derivatives or gradient, also known as the *Tangent Vector (TV)*, which is presented as follows:

$$TV = \begin{bmatrix} \dfrac{f(x_1 + \Delta x_1, \dots, x_{nc}) - f(x_1, \dots, x_{nc})}{|\Delta x_1|} \\ \vdots \\ \dfrac{f(x_1, \dots, x_{nc} + \Delta x_{nc}) - f(x_1, \dots, x_{nc})}{|\Delta x_{nc}|} \end{bmatrix} \quad (3)$$

where: *nc* is the number of control variables or actions; $f(x_1,..,x_{nc})$ is the objective function to be optimized; $x_1,\dots,x_{nc}$ are control variables; and Δx_k is a random and limited increment applied to x_k.

Under the AIS taxonomy, the set of control variables represents the Ab's, and the applied increments yields the hypermutated clones. Under the OPF taxonomy, the *TV* represents the system sensitivities, given a minor disturbance, around a certain operating point. These sensitivities are assessed for each control variable, and the final result defines a vector pointing to the most likely direction to achieve the optimization objective. Observe that if the objective function (or affinity) has an explicit mathematical expression, the *TV* can be substituted by the *Jacobian* vector (*JV*), which provides a more effective evaluation for the hypermutation process. The number of hypermutated clones must be equal to the number of control actions, i.e. *nc*, since this number is sufficient to ensure that the sensitivities of all space dimensions are being duly captured. In case the *JV* is being used, only one hypermutated clone is necessary, *nc =1*. Fig. 3 shows the adaptation of these concepts to the original AIS algorithm [22].

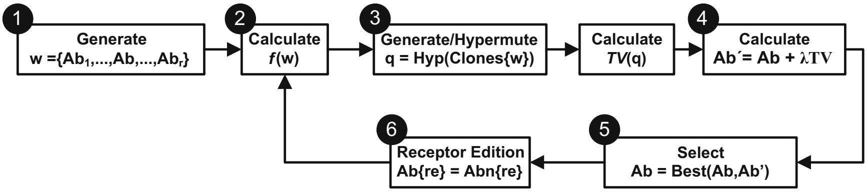


Fig. 3. Gradient-Based AIS Block Diagram

Each step or block of the previous diagram is detailed as follows:

1. Randomly choose a population w = $\{Ab_1,\dots,Ab_i,\dots,Ab_n\}$, with each individual defined as Ab_i = $\{x_1,\dots,x_j\dots,x_{nc}\}$, where nc represents the number of control variables or actions;
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 hypermutated clones q_i = $\{Ab_{i,1},\dots,Ab_{i,j}\dots,Ab_{i,nc}\}$ is generated, where $Ab_{i,j}$ = $\{x_{i,1},\dots, x_{i,j} + \Delta x_{i,j},\dots, x_{i,nc}\}$, where nc represents the number of control variables or actions, which are equal to the number of hypermutated clones. The hypermutated clones are then used to evaluate the TV_i according to eq. (3).
4. A new individual $Ab_i^{'}$ is assessed through eq. (4), where λ means a random step size:

$$Ab_i^{'} = Ab_i + \lambda \times TV_i\,; \tag{4}$$

5. Calculate the affinity of this new individual $Ab_i^{'}$ and check if it has a higher affinity compared to the original Ab_i; and if it does, the hypermutated clone takes its position in the population w;
6. The bests nb individuals among the original w 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 and it helps in searching for better solutions into different areas.

In the previous proposed algorithm, if the JV can be assessed, step 3 is slightly changed, and only the evaluation of the JV at Ab_i is necessary. Consequently, TV_i is substituted by JV_i in step 4. This new calculation process will further reduce the computing effort.

To evaluate the proposed algorithm, named as GbCLONALG, an optimization problem represented by equation 5, described in [22], is performed:

$$\text{Maximize } f(x_1,x_2) = x_1 \cdot sin(4\pi x_1) - x_2 \cdot sin(4\pi x_2 + \pi) + 1\,. \tag{5}$$

Figure 4 shows the solution achieved by the GbCLONALG considering the *TV* option and the simulation parameters defined by *Case 4* of Table 1. Table 1 shows the results of the proposed (GbCLONALG) and original (CLONALG) algorithms using different parameter simulations regarding: population size (Pop), number generations (Gen), and the number of fixed individuals (not selected for receptor editing). The outputs are the CPU time, in seconds (s), and the success rate, in percent (%), for a given configuration to find the global optimum. The number of clones was fixed in 2 for the GbCLONALG and round(0.1×Pop) for the CLONALG. The number of simulations to evaluate the success rate was 500. A Pentium 4 personal computer with 2.6 GHz is used for all simulations in a Matlab platform.

The simulations presented in Table 1 have demonstrated that the GbCLONALG reached better results considering both CPU time and success rate. For instance, using 60 Ab's and 10 generations (i.e. *Cases* 17-20), it is possible to achieve almost 100%

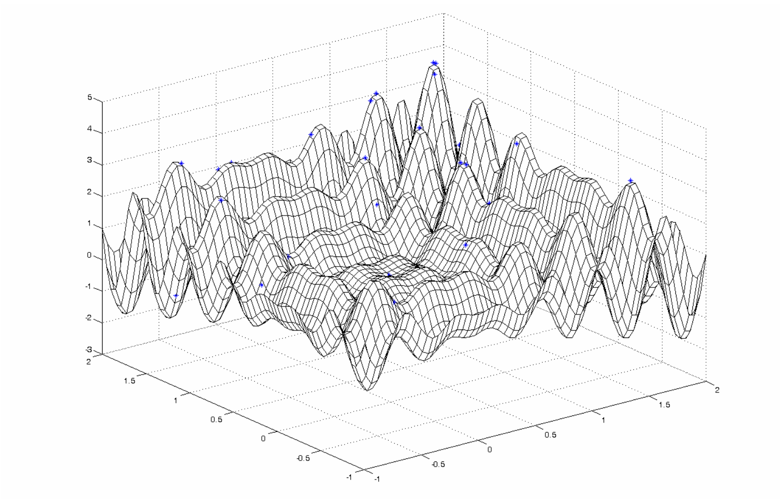


Fig. 4. Maximizing f(x_1,x_2) = $x_1$sin(4πx_1)-$x_2$sin(4πx_2+π) + 1: *Case 4*

Table 1. GbCLONALG and CLONALG Simulation Results

Case No.	Pop.	Gen.	Fixed Pop.	GbCLONALG		GLONALG	
				Success Rate (%)	Mean Time (s)	Success Rate (%)	Mean Time (s)
1	20	5	15	41	0.06	13	0.06
2	20	5	20	35	0.06	5	0.06
3	40	5	15	68	0.11	58	0.13
4	40	5	20	71	0.12	46	0.13
5	40	5	25	70	0.11	46	0.13
6	40	5	30	65	0.10	51	0.13
7	60	5	15	88	0.16	86	0.25
8	60	5	20	82	0.16	81	0.25
9	60	5	25	86	0.16	80	0.25
10	60	5	30	84	0.16	79	0.25
11	20	10	15	59	0.11	42	0.09
12	20	10	20	51	0.10	30	0.09
13	40	10	15	86	0.20	81	0.26
14	40	10	20	91	0.20	88	0.25
15	40	10	25	87	0.20	85	0.26
16	40	10	30	85	0.19	76	0.26
17	60	10	15	100	0.29	93	0.46
18	60	10	20	100	0.29	96	0.46
19	60	10	25	99	0.29	91	0.46
20	60	10	30	100	0.29	97	0.47

success rate using the GbCLONALG, with a CPU time of 0.27 seconds. Considering the same cases, the CLONALG has reached, in average, 94% of success rate, with a CPU time of 0.46 seconds. Observing the CPU time performance of both algorithms, as the number of population increases so do the differences, due to fact that the number of clones generated by the CLOANALG depends on the population size, while the GbCLONALG has it unchanged (i.e. nc=2). Moreover, GbCLONALG also achieved better success rate performance, as compared to the original CLONALG, due to the higher quality information captured by the *TV*. Finally, although the *TV* technique has been used in this simple application, the *JV* option would be much more appropriate since the analyzed function is well defined.

4 AIS Applied to Optimal Power Flow Problems

In order to apply the GbCLONALG to a typical OPF problem, the objective function f to be considered in this work will be the minimization of the total transmission losses, using shunt capacitor as control actions. This OPF problem can be mathematically described as follows:

$$Min\ f(V,\theta,b_{sh}) = \sum\nolimits_{(k,l \in L)} \left| P_{k,l} + P_{l,k} \right| \tag{6}$$

where: L represents the set of transmission equipment (i.e. lines and transformers); $P_{k,l}$ and $P_{l,k}$ are the active flow from bus k to bus l; b_{sh}, V and θ are vectors containing the shunt capacitors, voltage magnitudes and angles associated with the system buses.

The six-step algorithm, described in Section 3, is now applied to the OPF problem with some adaptation or reinterpretation. In Step 2, for instance, each individual (i.e. an Ab) represents a set of control actions x_c, also associated with the operating point represented by the state vector x_s. The value of the objective function f^{old} (i.e. Ab_i) is then calculated for this individual as follows:

$$f^{old} = f(x_s, x_c)\,. \tag{7}$$

In fact, equation 7 is to be performed for all individuals of the population. Now, in Step 3, hypermutated clones are generated according to equations (8) and (9):

$$g' = g(x_s, x'_c) \text{ with } x'_c = \{x_{c1},\ldots,x_{ci},\ldots,x_{cn}\} + \{0,\ldots,\Delta x_{ci},\ldots,0\}\,, \tag{8}$$

$$tv_i = (f^{new} - f^{old}) / \left|\Delta x_{ci}\right| \text{ with } f^{new} = f(x'_s, x'_c) \text{ and } x'_s = x_s - J^{-1} g' \tag{9}$$

where: g' is the corresponding vector of power injections for a given hypermutated clone x'_c; x'_s represents the consequences of this hypermutation process on the state vector; f^{new} (i.e. Ab'_i) is also the consequence of the whole hypermutation process regarding the value of the objective function; finally, tv_i is the i^{th} component of *TV*. Observe that equations (8) and (9) described a combination of the *TV* and *JV* concepts. The remaining steps (4, 5 and 6) follow the previously described GbCLONALG in Section 3.

The electric power network used to test the methodology was the standard IEEE 14-bus test system. A slight modification is carried out, where loads and generations are multiplied by a factor of 1.4, in order to stress the transmission system, and, consequently, to enhance system non-linearity. Figure 5 shows the used system, where the arrows indicate the consumers (i.e. loads). As discussed in reference [9], to install shunt capacitors in every bus in the system may cause several problems, and, therefore, it indicates buses 14, 11, 13 and 12 as the best locations for such controls.

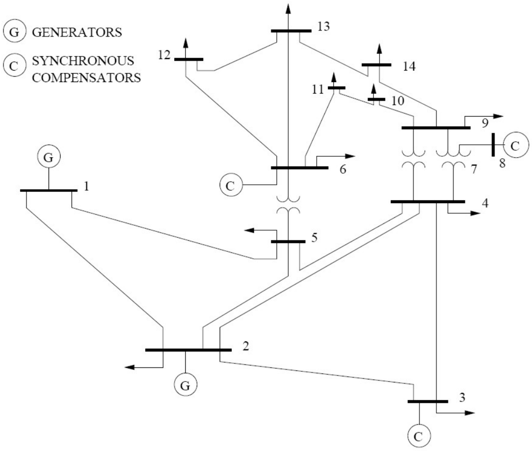


Fig. 5. IEEE 14-Bus Test System

Table 2 shows the results with both algorithms, i.e. GbCLONALG and the original CLONALG, considering different population sizes and number of generations. The output results are final transmission loss, in Megawatts (MW), and the CPU time in seconds (s). As it can be seen, the results demonstrated the very good performance achieved by proposed the GbCLONALG, which obtained speed-ups always greater than 2.

Table 2. GbCLONALG and CLONALG Simulation Results

Case No.	Pop.	Gen.	GbCLONALG		GLONALG	
			Final Loss (MW)	Time (s)	Final Loss (MW)	Time (s)
1	10	5	17.93	9	17.95	22
2	30	5	17.92	29	17.93	70
3	50	5	17.92	51	17.93	120
4	10	10	17.93	17	17.94	44
5	30	10	17.92	55	17.93	140
6	50	10	17.92	99	17.93	233
7	10	20	17.92	26	17.95	67
8	30	20	17.92	90	17.94	207
9	50	20	17.92	151	17.92	346

5 Conclusions

In this paper, a Gradient-Based Artificial Immune System algorithm has been proposed. It combines AIS and gradient concepts in order to improve both computing effort and searching robustness. The gradient numerical information, brought by the tangent vector and/or Jacobian, leads to a more efficient hypermutation process, and, consequently, to a faster approaching to local optima. The proposed GbCLONALG has been successfully implemented to minimize transmission system losses of a small standard test system. Other objective functions can be easily used following the same steps of the general proposed algorithm. Finally, the GbCLONALG is now being implemented in actual large-scale power networks, with thousands of variables being handled, in order to confirm the algorithm feasibility.

References

1. Granville, S.: Optimal Reactive Dispatch through Interior Point Methods. IEEE Transactions on Power Systems 19, 136–147 (1994)
2. Granville, S., Mello, J.C.O., Melo, A.C.G.: Application of Interior Point Methods to Power Flow Unsolvability. IEEE Transactions on Power Systems 11, 1096–1103 (1996)
3. Almeida, K.C., Salgado, R.: Optimal Power Flow Solutions under Variable Load Conditions. IEEE Trans. on Power Systems 15, 1204–1211 (2000)
4. Yoshida, H., Kawata, K., Fukuyama, Y., Takayama, S., Nakanishi, Y.: A Particle Swarm Optimization for Reactive Power and Voltage Control Considering Voltage Security Assessment. IEEE Transactions on Power Systems 15, 1232–1239 (2000)
5. Torres, G.L., Quintana, V.H.: On a Nonlinear Multiple-Centrality-Corrections Interior-Point Method for Optimal Power Flow. IEEE Transactions on Power Systems 16, 222–228 (2001)
6. Binato, S., Oliveira, G.C., Araújo, J.L.: A Greedy Randomized Adaptive Search Procedure for Transmission Expansion Planning. IEEE Transactions on Power Systems 16, 247–253 (2001)
7. Liu, M., Tso, S.K., Cheng, Y.: An Extended Nonlinear Primal-Dual Interior-Point Algorithm for Reactive-Power Optimization of Large-Scale Power Systems with Discrete Control Variables. IEEE Transactions on Power Systems 17, 982–991 (2002)
8. Shahidehpour, M., Yamin, H., Li, Z.: Market Operations in Electric Power System. J. Wiley & Sons, Inc. New York (1996)
9. Souza, A.C.Z., Honório, L.M., Torres, G.L., Lambert-Torres, G.: Increasing the Loadability of Power Systems through Optimal-Local-Control Actions. IEEE Transactions on Power Systems 19, 188–194 (2004)
10. Esmin, A.A.A., Lambert-Torres, G., Souza, A.C.Z.: A Hybrid Particle Swarm Optimization Applied to Loss Power Minimization. IEEE Transactions on Power Systems 20, 859–866 (2005)
11. Biskas, P.N., Ziogos, N.P., Tellidou, A., Zoumas, C.E., Bakirtzis, A.G., Petridis, V.: Comparison of Two Metaheuristics with Mathematical Programming Methods for the Solution of OPF. IEE Proceedings on Generation, Transmission and Distribution 153 (2006)
12. Leita da Silva, A.M., Manso, L.A.F., Mello, J.C.O, Billinton, R.: Pseudo-Chronological Simulation for Composite Reliability Analysis with Time Varying Loads. IEEE Transactions on Power Systems 15, 73–80 (2000)

13. Leite da Silva, A.M., Resende, L.C., Manso, Miranda, V.: Composite Reliability Assessment Based on Monte Carlo Simulation and Artificial Neural Networks. IEEE Transactions on Power Systems, Accepted for publication. IEEE, New York (2007)
14. Ishak, S., Abidin, A.F., Rahman, T.K.A.: Static Var Compensator Planning Using Artificial Immune System for Loss Minimisation and Voltage Improvement. In: National Power and Energy Conference, PECon, pp. 41–45. IEEE, Malaysia (2004)
15. Rahman, T.K.A., Suliman, S.L., Musirin, I.: Artificial Immune-Based Optimization for Solving Economic Dispatch in Power System. In: Apolloni, B., Marinaro, M., Nicosia, G., Tagliaferri, R. (eds.) WIRN 2005 and NAIS 2005. LNCS, vol. 3931, pp. 338–345. Springer, Heidelberg (2006)
16. Liao, G.C: Application of an Immune Algorithm to the Short-Term Unit Commitment Problem in Power System Operation. IEE Proceedings-Generation Transmission and Distribution 153, 309–320 (2006)
17. Russel, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall, New York (1995)
18. Pelikan, M., Goldberg, D.E., Lobo, F.G.: A Survey of Optimization by Building and Using Probabilistic Models. Computational Optimization and Applications. Kluwer Academic Publishers, The Netherlands (2002)
19. Sheng, W., Swift, S., Zhang, L., Liu, X.: A Weighted Sum Validity Function for Clustering With a Hybrid Niching Genetic Algorithm. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics 35, 1156–1167 (2005)
20. Hunt, J.E., Cooke, D.E.: Learning Using an Artificial Immune System. Journal of Network and Computer Applications 19, 189–212 (1996)
21. Castro, L.N., Zubben, F.J.V.: Artificial Immune Systems: Part I – Basic Theory and Applications. Technical Report, FEEC/UNICAMP. Campinas, Brazil (2000), [Online]. Available: http:www.dca.fee.unicamp.br/ lnunes/immune.html
22. Castro, L.N., Zubben, F.J.V.: Learning and Optimization Using the Clonal Selection Principle. IEEE Transactions on Evolutionary Computation 6, 239–251 (2002)
23. Carpaneto, E., Cavallero, C., Freschi, F., Repetto, M.: Immune Procedure for Optimal Scheduling of Complex Energy Systems. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 309–320. Springer, Heidelberg (2006)
24. Kundur, P.: Power System Stability and Control. McGraw-Hill, New York (1994)
25. Wood, A.J., Wollenberg, B.F.: Power Generation Operation and Control, 2nd edn. J. Wiley & Sons, Inc. New York (1996)

Multimodal Dynamic Optimization: From Evolutionary Algorithms to Artificial Immune Systems

Nikolaos Nanas and Anne De Roeck

Computing Department, The Open University, Milton Keynes, MK7 6AA, U.K.
{N.Nanas,A.deRoeck}@open.ac.uk

Abstract. Multimodal Dynamic Optimisation is a challenging problem, used in this paper as a framework for the qualitative comparison between Evolutionary Algorithms and Artificial Immune Systems. It is argued that while Evolutionary Algorithms have inherent diversity problems that do not allow them to successfully deal with multimodal dynamic optimisation, the biological immune system involves natural processes for maintaining and boosting diversity and thus serves well as a metaphor for tackling this problem. We review the basic evolutionary and immune-inspired approaches to multimodal dynamic optimisation, we identify correspondences and differences and point out essential computation elements.

1 Introduction

The domain of Artificial Immune Systems (AIS) has only recently emerged within the broader discipline of Bio-inspired Computing. There have been doubts on the necessity of yet another biologically inspired approach, especially given the perceived similarity to the well established Evolutionary Algorithms (EAs). Both use a population of simple representational units, antibodies and chromosomes respectively, that are matched/evaluated against a certain antigenic/solution space. Other than introducing a new set of theories and concepts from Immunology in Bio-inspired Computing, AIS have been treated as another algorithmic variation of Evolutionary Algorithms.

With this paper we wish to contribute to the argument stating that despite some common computational elements, AIS can be fundamentally different to EAs. We will use Multimodal Dynamic Optimisation (MDO) as a framework for comparing the two approaches at the algorithmic level to highlight their essential differences. In MDO there is no single and static optimum, but instead, a varied number of optima that continuously change their position and shape in the solution space. Changes can be both local or global, modest or radical, slow or fast, but they are not chaotic. MDO poses challenging requirements that are going to be identified and used to characterise and compare EAs and AIS.

Although life and many real world problems are inherently dynamic, research in Evolutionary Computation, focuses traditionally on optimisation of static problems, or problems treated as static, with only one global optimum. Recently

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 13–24, 2007.

there has been a growing interest for tackling non-stationary problems with rugged landscapes, where the various optima have to be found and tracked over time. It has already been recognised, that standard EAs suffer when applied to MDO, because they eventually converge to one optimum and loose the diversity necessary for covering multiple peaks and for adapting to changes in them.

The literature on Evolutionary Computation is already rife with methods for making EAs comply with the requirements of MDO. A comprehensive review of existing evolutionary approaches to MDO is outside the scope of this paper. We will concentrate on the basic techniques and will argue that these are at large ad hoc remedies without always biological foundations. It will be put forward that the drawbacks of EAs in the case of MDO derive from their fundamental algorithmic components and so existing remedies are only patches to intrinsic diversity problems. Furthermore, the proposed remedies point explicitly or implicitly towards the direction of AIS. The immune system deals with a varied and constantly changing pathogenic environment from within a developing organism. As such, it has inbuilt mechanisms for preserving and boosting the diversity of the immune repertoire. In AIS these mechanisms are translated into algorithmic processes that control a system's diversity and allow it to deal naturally with MDO. While there is some correspondence with the effects achieved with algorithmic remedies in the case of EAs, in AIS the diversity controlling mechanisms are both inherent and biologically grounded.

In the rest of this paper we first set out the requirements of MDO. We then use these requirements to critically review evolutionary (sec. 3) and immune-inspired approaches (sec. 4) to MDO. Finally we discuss their similarities and differences and point out directions for future work.

2 Requirements for Multimodal Dynamic Optimisation

A generalised MDO problem can be expressed as a multidimensional space where specific regions correspond to the current solutions, or collectively to an overall solution to a problem. The underlying problem is dynamic, but not chaotic, and hence its solutions constantly change over time. Therefore, it is at least impractical to treat the problem's solutions as points in the multidimensional space, since any point in the space only momentarily becomes a local, or global optimum. It is more appropriate to consider solution regions that change in shape, area and position due to the problem's dynamics. The latter can have different scales. At the lower level, short-term dynamics moves a solution region locally or modifies its shape. At a higher level, long-term dynamics causes new solution regions to emerge and others to disappear. In the extreme case, the space itself can have dynamics that modify its dimensions.

An example problem that demonstrates the characteristics of MDO is Adaptive Information Filtering (AIF). AIF seeks to provide a user with information relevant to the user's multiple and changing interests. Here the user's interests define regions in a multidimensional information space. Changes in the user's interests can be short-term variations in the level of interest in certain topics, or even more radical long-term changes like the emergence of a new topic of interest, or loss of interest in a certain topic.

To tackle MDO a system has to be able to trace, represent and track the solution regions. This is a challenging task with specific requirements. As already mentioned, both EAs and AIS use populations of simple units that are typically expressed according to the dimensions of the underlying space. *Diversity* – i.e. the presence of a wide range of variation in the qualities or attributes of the individuals involved in a population – is crucial for biological processes like evolution and immunity. Regarding MDO, diversity ensures that the population covers the various solution regions and remains spread around the space in anticipation of changes in them. The diversity problems of EAs have already been recognised and as we will further discuss in the next section, most proposed remedies focus on administering diversity into the population. There are however additional requirements for MDO that have been generally ignored.

As the number and area of solution regions varies in time the number of individuals that are required to represent them should vary accordingly. For example, in the case of AIF, whenever new topics of interest emerge additional individuals are needed to cover the new interesting areas in the information space. In contrast, when the user looses interest in a specific topic the corresponding individuals are no longer required, or it is at least inefficient to maintain them in the population. The cardinality of the population should be adjusted according to the problem characteristics.

Finally, the algorithmic processes should reflect the problem's dynamics. This implies that each individual should be able to move locally, or that the interactions and relative importance of individuals can be modified. At the global level new individuals are required to cover emerging solution regions and no longer competent individuals should be removed.

3 EAs and MDO

The insight behind EAs is that the fundamental components of biological evolution can be used to evolve solutions to problems within computers. They are stochastic search techniques that have been traditionally applied to optimisation problems. A GA starts with a problem and a *population* of random solutions. Each individual in the population, is called a *chromosome*, and is represented as a string over a finite alphabet. The population evolves through iterations, called *generations.* At each generation, each chromosome in the current population is evaluated using some appropriate *fitness function*, which measures how good the solution embodied in a chromosome is. Chromosomes are selected for reproduction according to their relative fitness. Two chromosomes mate using the *crossover* operation that combines parts of each parent chromosome to produce possibly better offspring that replace the less fit individuals in the current population. Occasionally, to randomly explore the solution space, chromosomes are slightly modified with the *mutation* operation. The evolutionary process is iterated until a certain condition is met. In static optimisation problems for example, the iterations terminate once the observed improvements drop below a certain level.

EAs suffer in the case of MDO because they tend to converge to a single optimum. They progressively lose diversity as the optimum proliferates and spreads over the population. The remedies proposed in the literature attempt to overcome this drawback by artificially maintaining and/or injecting diversity in the population. Four types of remedies have been identified [1]:

1. explicit action is taken to increase diversity whenever changes are detected.
2. convergence is avoided by spreading out the population.
3. the GA is supplied with memory of past chromosomes that are reused whenever the optimum returns to a previous location.
4. multiple subpopulations are used to represent the solution regions and to search for new ones.

The most straightforward way for increasing the diversity of a population in EAs is through mutation, which places new offspring randomly in the solution space. Of course the effect of high mutation rates on adaptation can be unpredictable and should be controlled. One solution is to drastically increase the mutation rate only after a change in the solution space has been detected [2]. Alternatively, the range of mutation can be adjusted according to the detected changes [3].

To avoid convergence and maintain diversity in the population one has to assure that the population remains spread through out the solution space. In [4], the author describes the *random immigrants* method that in every generation replaces part of the population with randomly generated individuals, which are not the product of the genetic operations. Another solution is to take into account the similarity between individuals during selection and replacement. For example in [5], the first parent is selected based on its fitness and the second parent according to its similarity with the first. Furthermore, a method called “worst among most similar” is employed to identify the individual to be replaced by the new offspring. In [6], “direct replacement” is proposed, which avoids convergence by enforcing direct replacement of parents by their offspring after crossover.

Supplying a GA with memory has been suggested as a way of dealing with dynamic environments, especially in those cases where there is a fixed number of optima, or the optima tend to return to the same positions in the solution space. Memory can be either explicit or implicit. In the first case, individuals from past generations are externally stored and reused whenever it is deemed necessary. *Elitist strategies* that periodically store the best individuals in the population for future use belong to this category. For instance, [7] extends the random immigrants approach described above using an elitist strategy. The best individuals from past generations are stored and used to generate the immigrants. A similar approach to the open shop scheduling problem is described in [8]. How to best store and reuse individuals has not been yet clarified. [9] compares various strategies for storing and retrieving individuals from the external memory and his experiments indicate that memory is only advantageous when the optimum repeatedly returns to the exact previous location and not to a slightly different one.

Rather than being external, implicit memory is encoded in the genotype itself. Diploidy is a characteristic example of implicit memory. Each individual has a pair of chromosomes and a separation exists between genotype and phenotype. During evaluation an individual's diploid genotype is first mapped into a haploid phenotype by some dominance mechanism. The dominance mechanism can be modified in response to changes in the environment, e.g. by reversing the dominance relation [10]. Nevertheless, experiments show that a simple haploid GA with an appropriate mutation rate performs comparably to the diploidy approach. More importantly, the diploid mechanism is able to learn a problem with two changing optima, but fails completely in the case of more that two optima.

Instead of memory that becomes obsolete if the environment does not change in such a way that specific solution regions are revisited, another remedy is to maintain subpopulations that cover the solution regions and track their changes. In [11] the "self-organising-scouts" approach is described, which divides the search space into regions and assigns a small scout subpopulation whenever a new promising region is detected. Similarly, the "multinational" GA structures the population into subpopulations ("nations") based on a method for detecting valleys in the solution space [3, 12]. Selection is performed at both national and global level and in the latter case special care is taken so that nations can not easily go extinct. The algorithm also includes processes for migration to new promising regions and for merging nations that converge to the same region. A similar approach, called the "shifting balance GA", which maintains a "core" population and "colonies" exploring different regions in the environment is described in [13].

The above is only a sample of the techniques that appear in the literature for tackling MDO with EAs. For a more comprehensive review see [1] and the proceedings of the workshop on Evolutionary Algorithms for Dynamic Optimization. MDO is clearly a significant problem that has attracted a lot of interest, yet, the proposed techniques are to a great extent ad hoc remedies without clear biological correspondence. To our knowledge there is no biological evidence showing that mutation rate can vary based on changes in the environment[1], or that it can be encoded in the genome itself. It is also unclear how evolution could produce individuals at specific, non random, phylogenetic regions. The lack of biological correspondence is further reflected in the terminology used to describe these techniques, which adopts concepts from sociology and politics, but without being firmly grounded in some specific theory. Of course, from an engineering point of view, moving away from biological metaphors is not necessarily inadequate, but nevertheless it hinders the constructive interplay between computing and biology. Research in evolutionary computation should look more thoroughly into the biological evolutionary processes that produce extremely diverse species.

Moreover, we argue that the problems EAs face in the case of MDO are intrinsic in nature and the suggested remedies are treating the symptoms rather than the source of the problem. In particular, we believe that the diversity problems of EAs derive from the way the principle of the "survival of the fittest" is implemented. According to evolutionary theory the fittest an organism is, the higher

[1] Other than changes in radiation levels.

the probability of reproduction. This is typically translated into an algorithmic process for choosing parents to mate according to their **relative** fitness. For example, in [14] a fixed percentage of the fittest individuals is selected. To more accurately reflect natural evolution, one may use *roulette wheel* selection that assigns to each individual reproduction probability proportional to its fitness [15]. As a consequence of selection based on relative fitness, individuals covering the region of the space that corresponds to the best current solution are more likely to mate and reproduce. Crossover, combines the parent chromosomes to produce offspring with genotypes made out of genes inherited from the parents. The offspring are thus bound to regions in proximity to their parents. Since a developmental process capable of inducing large phenotypic differences out of small genotypic differences is typically excluded, the crossover operation can not generate diversified offspring. It is up to mutation, which randomly introduces diversity in the genotype of offspring, to produce radical new solutions in new previously unexplored space regions. Even so, in multidimensional spaces, only mutation with a high rate affecting multiple loci in the genotype, could generate diversified enough offspring. Last but not least, the offspring replace the less fit individuals in the population. In other words, it is typically assumed that the population has a certain fixed capacity, predefined as a system parameter. Combined with selection based on relative fitness, this leads to the spread of the best individuals in the population, and the progressive loss of diversity. Individuals covering the best current solution region multiply at the cost of individuals covering other, possibly promising or even essential, solution regions. A fixed population capacity, does not comply with the requirements of the MDO problem. The population's cardinality should reflect the current problem needs: in spaces with a few solution regions less individuals are required[2] and more individuals are necessary for spaces with multiple solution regions. There is a lack in EAs of a mechanism that controls the size of the population in response to changes in the environment.

Overall, diversity problems in EAs derive from their basic algorithmic processes and their interaction. Selection based on relative fitness causes the fittest individuals to proliferate, creating similar offspring that can only be diversified with intense mutation, and which replace the least fit, but possibly important individuals in the population. The reviewed remedies are external processes that treat the effects of the intrinsic diversity problems of EAs. They unnecessarily increase the parameters involved. For example in the Multinational GA [12], the parameters involved include in addition to the population size, mutation probability and crossover probabilities, the ratio between global and national selection, and mutation variance for individuals close to and distant from the nation's centroid. In dynamic environments however, the number of system parameters should be minimised because it is difficult to tune them as they most likely change along with the landscape. How to best control the parameters of EAs during adaptation and according to the problem is a research question in its own right [16]. One solution is to encode the additional parameters in the

[2] Or can be sustained.

genome, but experimental results show that this causes a lag in the algorithms ability to keep up with changes in the environment [12].

4 AIS and MDO

AIS are not meant to be accurate models of the biological immune system, but use relevant processes and concepts. Common components of AIS include: a *representation* of the cell (e.g. antibody) structure, a measure of the *affinity* between cells and an *algorithm* that simulates immune processes such as negative selection, clonal selection and idiotypic effects. Currently, various AIS exist with many and diverse applications[3].

The biological immune system serves well as a computational metaphor for tackling MDO, due to its ability to maintain and boost diversity. This is achieved in two ways: with *heterostasis*, the preservation of diversity and *heterogenesis*, the introduction of diversity. On one hand, the diversity of the immune repertoire is preserved due to the way immune cells (lymphocytes in particular) are triggered to clone. According to *clonal selection*, a lymphocyte is triggered to clone when its antibodies recognise an antigen. The cloning rate is analogous to the affinity between antibody and antigen and is usually high. This is a local selection scheme that chooses cells to proliferate not according to their relative fitness, but rather according to the absolute value of their affinity to an invading antigen. So even lymphocytes that have not been successful so far in recognising antigens can be triggered to clone when the antigenic environment changes and new matching antigens invade the organism. In addition, long-lived memory cells that have been successful in the past and idiotypic effects [19], due to the ability of antibodies to recognise each other and form chains of suppression and reinforcement, contribute further to heterostasis.

Heterogenesis on the other hand is achieved in two ways. When a lymphocyte is triggered to clone the cloning process is not perfect, but is subjected to intense mutation, called *somatic hypermutation*, that results in slightly different lymphocytes and hence antibodies, which can be a better match to the invading antigen. Further diversity is introduced with the recruitment of new lymphocytes manufactured in the bone marrow. Hypermutation and cell recruitment cause new antibody types to be added to the immune repertoire, and play the role of local and global search respectively. Finally, it should be noted that the clones and the recruited cells do not necessarily replace existing immune cells. Individual cells die through natural decay processes (like *apoptosis*), independently of how successful they have been in recognising antigens. Hence, the size of the immune repertoire is not fixed. This implies that some mechanism exists for controlling repertoire size. For example, in *self-assertion* models of the immune antibody network, *endogenous selection*, according to which the network itself selects which recruited cells are going to survive in the repertoire, plays this role. Using a computational self-assertion model, De Boer and Perelson have shown that the immune system can control both its size and connectivity [20].

[3] Textbooks in AIS include [17, 18].

Overall, by combining heterostatis with heterogenesis, and more specifically, through *affinity maturation* (i.e. the combined effect of clonal selection and hypermutation), memory, idiotypic effects, cell death and recruitment, the immune system succeeds in maintaining and boosting its diversity. Compared with EAs, this is a significant property that invites the application of AIS to MDO. AIS have already been applied to optimisation problems (see [21] for a survery). Here we concentrate on immune-inspired approaches to MDO in particular.

The ability of immune processes to maintain and boost diversity has been recognised by researchers in evolutionary computation. Many hybrid algorithms that introduce immune inspired elements in EAs have been proposed. In [22], the authors describe the Immunity based Genetic Algorithm, which incorporates in addition to genetic operations, clonal selection and idiotypic effects. [23] proposed another hybrid that includes clonal selection and somatic hypermutation performed using gene libraries. A secondary memory population stores individuals that have proved successful in the past and redeploys them whenever a degradation in performance is detected. These hybrids are only partial, engineered solutions that do not fully exploit the computational capabilities of the immune system. They typically exclude a local selection schema and use a fixed population size.

The Simple Artificial Immune System (SAIS) [24] was one of the first attempts to fully exploit immunity as a metaphor for tackling dynamic optimisation. SAIS encodes cells as binary strings, and comprises clonal selection, cell decay, recruitment and idiotypic effects. Although K-tournament selection on the best than average cells is adopted rather than a local selection schema, the repertoire size is not fixed, but is controlled through recruitment, idiotypic effects and cell decay. In comparative experiments on changing environments with a single optimum, SAIS demonstrated improved reactiveness and robustness in comparison with EAs. However, SAIS performed worse than a GA with Lamarckian local search and had unstable dynamics. This latter finding is representative of the difficulty involved in devising a mechanism for appropriately controlling the number of immune cells in an adaptive manner. To deal with this issue, in [25] the authors simplify the system's dynamics by dropping idiotypic effects and instead grouping cells into a predefined number of gatherings that play the role of diversity preserving memory. They found however, that the system became sensitive to the number of gatherings.

opt-aiNet is a multimodal function optimisation algorithm that exhibits dynamic allocation of repertoire size [26]. It comprises affinity maturation, recruitment of new random cells and idiotypic effects that cause similar cells to be removed. A modified version, called dopt-aiNet, that deals with dynamic environments is described in [27]. It extends opt-aiNet with a separate memory population, a search procedure for controlling decay, new mutation operators and a different measure of affinity. It also includes a maximum population size for avoiding the unnecessary escalation in the number of cells in opt-aiNet, when multiple optima have to be captured. According to the authors these additions improve opt-aiNet's ability to quickly find optimal solutions and to maintain diversity when dealing with dynamic environments.

Nootropia is an immune-inspired model applied to adaptive information filtering [28, 29]. It is a self-assertion model that follows Varela's autopoietic view of the immune system [30]. Rather than being antigen driven it uses a self-organising network of antibodies to define and preserve the organism's identity, which in the case of adaptive information filtering corresponds to a user's multiple interests. Antibodies correspond to features describing information items and the affinity between antibodies is statistically measured. It has been argued and supported experimentally that the nonlinearity that results from the interaction of antibodies is beneficiary for capturing multiple regions of interest in the information space [31]. Adaptation is achieved through a process of self-organisation that renders the system open to its environment. New antibody types produced by the bone marrow are recruited and incompetent antibodies are eventually removed from the repertoire. The combination of dynamics that reinforces and suppresses existing antibody types and metadynamics that controls recruitment and removal of cells, allow the system to drift, constantly following changes in user interests. Experiments have demonstrated Nootropia's ability to adapt to a variety of interest changes [29], while yet unpublished results show that, in accordance with [20], the system can control both its size and connectivity.

5 Discussion and Future Directions

MDO is a challenging problem and how to best deal with it remains an open research question. It has already been recognised that EAs face diversity problems when dealing with MDO and various ad hoc remedies for artificially maintaining or injecting diversity have been proposed in the literature. However, we put forward that the diversity problem's of EAs are intrinsic in nature. They are mainly due to the combined effect of selecting parents for reproduction based on their relative fitness and using a fixed population size, which implies that offspring replace existing individuals and typically the less fit. On the contrary, the biological immune system involves natural processes for maintaining and boosting diversity. It exhibits both dynamics at the level of individual cell concentration and metadynamics at the level of antibody types. These control the repertoire's diversity, but also its size, and allow the immune system to adapt both locally and globally to changes in the antigenic space. Immune-inspired algorithmic elements have been used to hybridise EAs for MDO, but there are also fully fledged AIS for tackling the MDO problem.

There is clearly a plenitude of approaches, but no agreement yet on which one is the best and why. There have been experiments showing the benefits of maintaining and boosting diversity for MDO, but they don't compare the different mechanisms for doing so. Furthermore, the terminology used is either just descriptive (e.g. "random immigrants"), or reflects the biological phenomenon being modelled (e.g. "fitness" and "affinity"), rather than the underlying computational elements, thus obscuring essential similarities or differences. As already proposed in [32], a common computational framework for making correspondences and transferring mathematical results across disciplinary boundaries is required. Before any performance comparisons between different approaches to MDO, common computational elements should be distilled. For example, there appears to be a correspondence between the remedies used to overcome the diversity problems of EAs (section 3) and immune processes for maintaining and boosting diversity (section 4). Introducing diversity by increasing the rate of mutation in EAs, is achieved with hypermutation in AIS. Adding random individuals ("immigrants") is analogous to the generation of new lymphocytes by the bone marrow based on gene libraries. Suppression of similar individuals, which is accomplished in EAs with ad hoc methods like replacing the "worst among most similar", is the effect of idiotypic interactions between antibodies in AIS. Memory in AIS is either achieved explicitly with long-lived cells, or implicitly due to idiotypic effects. Subpopulations ("nations") of cells can also emerge in AIS due to interactions between antibodies. Without doubt there are common computational elements despite differences in describing them, but there are also essential differences in the way these elements are implemented and combined.

A local selection schema, rather than selection based on relative fitness, was deemed essential for maintaining diversity. Interactions between individuals function as another diversity controlling mechanism through suppression and excitation. In self-assertion models they also affect the recruitment and removal of individuals. Furthermore, they give rise to non-linear (developmental) processes that can be crucial for adaptation. A dynamically controlled population/repertoire size that reflects the environment's characteristics is also essential, but has only been adopted by a fraction of the reviewed approaches. Dynamically controlling the size of the population/repertoire is still an open research question. It is also yet unclear how to best track changes in the environment, which requires an appropriate combination of dynamics for local and metadynamics for global adaptation. To tackle these issues, we should pursue an abstraction of the various approaches according to these fundamental computational elements and a mapping from computational elements to the requirements of MDO. We will then be able to draw justifiable conclusions about the behaviour of different systems and how it is affected by their elementary processes. We further expect that such an analysis will raise questions about the underlying biological processes like: What is the role of development in evolution? Is evolution really a competitive process of "survival of the fittest", or the result of the ongoing interaction between adaptive individuals in changing environments?

Finally, we believe that existing experimental settings do not fully reflect the requirements of MDO (section 2). The common practice is to use simulated experiments like pattern tracking [24], dynamic gaussian [4] and dynamic knapsack problems [23]. These experiments have the advantage of being controlled and reproducible, but in most cases they lack in complexity, since they are based on low dimensional spaces. More importantly, changes in the environment are simulated by periodically relocating one or more optima. Changes are thus discrete rather than continuous and the evaluated systems have to be able to re-converge to the new positions of the optima, instead of being required to constantly track changes in time. MDO is simplified to the task of reinitialising the convergence process every time the environment changes, starting with a population that is hopefully better than random. There is a need for controlled and reproducible experimental settings that simulate continuous changes in multidimensional spaces. These could be based on real world data like the genomics data of the Text Retrieval Conference (TREC)[4] and the data for the International Challenge to create and train computationally intelligent algorithms on clinical free-text[5]. The experimental methodology described in [29] is a first step towards this direction. MDO can form the experimental test bed for the comprehensive analysis of EAs, AIS and possibly other biologically-inspired algorithms and for further constructive interaction between biology and computing.

References

1. Branke, J.: Evolutionary Optimization in Dynamic Environments. Kluwer Academic Publishers, Dordrecht (2001)
2. Cobb, H.G.: An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments. Technical Report 6760 (NLR Memorandum). Washington, D.C (1990)
3. Vavak, F., Jukes, K., Fogarty, T.C.: Learning the local search range for genetic optimisation in nonstationary environments. In: IEEE Int. Conf. on Evolutionary Computation (ICEC'97), pp. 355–360. IEEE, New York (1997)
4. Grefenstette, J.J.: Genetic algorithms for changing environments. In: 2nd Int. Conf. on Parallel Problem Solving from Nature, pp. 137–144 (1992)
5. Cedeno, W., Vemuri, V.R.: On the use of niching for dynamic landscapes. In: IEEE Int. Conf. on Evolutionary Computation (ICEC'97), pp. 361–366. IEEE Computer Society Press, Los Alamitos (1997)
6. Rowe, J., East, I.R.: Direct replacement: A genetic algorithm without mutation which avoids deception. In: Evo Workshops, pp. 41–48 (1994)
7. Yang, S.: Memory-based immigrants for genetic algorithms in dynamic environments. In: Proc. of (GECCO'05), pp. 1115–1122. ACM Press, New York (2005)
8. Louis, S.J., Xu, Z.: Genetic algorithms for open shop scheduling and re-scheduling. In: 11th Int. Conf. on Computers and their Applications (ISCA), pp. 99–102 (1996)
9. Branke, J.: Memory enhanced evolutionary algorithms for changing optimization problems. In: Proc. of the (CEC'99), vol. 3, pp. 1875–1882. IEEE Press, New York (1999)
10. Lewis, J., Hart, E., Ritchie, G.: A comparison of dominance mechanisms and simple mutation on non-stationary problems. In: Proc. of (PPSN V), pp. 139–148 (1998)
11. Branke, J., Kau, T., Schmidt, L., Schmeck, H.: A multi-population approach to dynamic optimization problems. In: 4th International Conference on Adaptive Computing in Design and Manufacture (ACDM 2000) (2000)
12. Ursem, R.K.: Multinational GAs: Multimodal optimization techniques in dynamic environments. In: Proc. of (GECCO-2000), pp. 19–26. Morgan Kaufmann, San Francisco (2000)

[4] http://ir.ohsu.edu/genomics/data.html

[5] http://www.computationalmedicine.org/challenge/index.php

13. Oppacher, F., Wineberg, M.: The shifting balance genetic algorithm: Improving the GA in a dynamic environment. In: Proc. of (GECCO-99), pp. 504–510. Morgan Kaufman, San Francisco (1999)
14. Sheth, B.D.: A Learning Approach to Personalized Information Filtering. Master of Science, Massachusetts Institute of Technology (1994)
15. Winiwarter, W.: PEA - a personal email assistant with evolutionary adaptation. International Journal of Information Technology 5 (1999)
16. Eiben, A.E., Hinterding, R., Michalewicz, Z.: Parameter control in evolutionary algorithms. IEEE Trans. on Evolutionary Computation 3, 124–141 (1999)
17. Dasgupta, D. (ed.): Artificial Immune Systems and Their Applications. Springer, Heidelberg (1998)
18. Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, Heidelberg (2002)
19. Jerne, N.K.: Towards a network theory of the immune system. Annals of Immunology 125, 373–389 (1973)
20. De Boer, R.J., Perelson, A.S.: Size and connectivity as emergent properties of a developing immune network. Journal of Theoretical Biology 149, 381–424 (1991)
21. Wang, X., Gao, X.Z., Ovaska, S.J.: Artificial immune optimization methods and applications–a survey. IEEE Int. Conf. on Systems, Man and Cybernetics 4, 3415–3420 (2004)
22. Tazawa, I., Koakutsu, S., Hirata, H.: An immunity based genetic algorithm and its application to the VLSI floorplan design problem. In: IEEE Int. Conf. on Evolutionary Computation (ICEC'96), pp. 417–421. IEEE Computer Society Press, Los Alamitos (1996)
23. Simões, A.B., Costa, E.: An immune system-based genetic algorithm to deal with dynamic environments: Diversity and memory. In: 6th International Conference on Neural Networks and Genetic Algorithms (ICANNGA'03), pp. 168–174 (2003)
24. Gaspar, A., Collard, P.: From GAs to artificial immune systems: Improving adaptation in time dependent optimization. In: Proceedings of the Congress on Evolutionary Computation, vol. 3, pp. 1859–1866. IEEE Press, Los Alamitos (1999)
25. Gaspar, A., Collard, P.: Two models of immunization for time dependent optimization. IEEE Int. Conf. on Systems, Man and Cybernetics 1, 113–118 (2000)
26. de Castro, L., Timmis, J.: An artificial immune network for multimodal optimisation. In: Congress on Evolutionary Computation, pp. 699–704. IEEE, New York (2002)
27. de França, F.O, Zuben, F.J.V., de Castro, L.N.: An artificial immune network for multimodal function optimization on dynamic environments. In: Proc. of (GECCO-2005), pp. 289–296. ACM Press, New York (2005)
28. Nanas, N., Uren, V., De Roeck, A.: Nootropia: a user profiling model based on a self-organising term network. In: 3rd International Conference on Artificial Immune Systems, pp. 146–160 (2004)
29. Nanas, N., Uren, V., De Roeck, A.: Immune-inspired adaptive information filtering. In: 5th International Conference on Artificial Immune Systems, pp. 418–431 (2006)
30. Varela, F.J., Coutinho, A.: Second generation immune network. Immunology Today 12, 159–166 (1991)
31. Nanas, N., Uren, V., De Roeck, A., Domingue, J.: Multi-topic information filtering with a single user profile. In: 3rd Hellenic Conference on Artificial Intelligence, pp. 400–409 (2004)
32. 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, 315–338 (2005)

NAIS: A Calibrated Immune Inspired Algorithm to Solve Binary Constraint Satisfaction Problems*

Marcos Zuñiga[1], María-Cristina Riff[2], and Elizabeth Montero[2]

[1] Projet ORION, INRIA Sophia-Antipolis
Nice, France
Marcos.Zuniga@sophia.inria.fr

[2] Department of Computer Science, Universidad Técnica Federico Santa María,
Valparaíso, Chile
{María-Cristina.Riff,Elizabeth.Montero}@inf.utfsm.cl

Abstract. We propose in this paper an artificial immune system to solve CSPs. The algorithm has been designed following the framework proposed by de Castro and Timmis. We have calibrated our algorithm using Relevance Estimation and Value Calibration (REVAC), that is a new technique, recently introduced to find the parameter values for evolutionary algorithms. The tests were carried out using random generated binary constraint satisfaction problems on the transition phase where are the hardest problems. The algorithm shown to be able to find quickly good quality solutions.

1 Introduction

Constraint satisfaction problems (CSPs) involve finding values for problem variables subject to constraints on which combinations are acceptable. Over the few years, many algorithms and heuristics were developed to find a solution of CSPs. Following these trends from the constraint research community in the bio-inspired computation community, some approaches have also been proposed to tackle CSP with success such that evolutionary algorithms [4], [8], [10], [12], [11], [14], ants algorithms [13]. Given that recent publications indicate that artificial immune systems offer advantages in solving complex problems [1], [3], our goal here is to propose an efficient immune inspired algorithm which can solve CSPs. Immune artificial systems as well as evolutionary algorithms are very sensitive to the values of their parameters. Garret in [18] proposed a parameter-free clonal selection using adaptive changes.

In this paper, we focalize our attention in a new method proposed for tuning, that is a method that uses statistical properties to determine the best set of parameter values for an evoltuionary algorithm.
The contributions of this paper are:

* Supported by Fondecyt Project 1060377.

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 25–34, 2007.

- An immune inspired algorithm which can solve hard CSPs,
- A new application of the tuning method Relevance Estimation and Value Calibration (REVAC) proposed for evolutionary algorithms, [15],

The paper is structured as follows. In the next section, we define the Constraint Satisfaction Problem. In section 3 we introduce our new approach NAIS. The results of tests and a comparison with other incomplete method are given in section 4. In our summary, we give some conclusions and future works.

2 Binary Constraint Satisfaction Problems

For simplicity we restrict our attention here to binary CSPs, where the constraints involve two variables. Binary constraints are binary relations. If a variable i has a domain of potential values D_i and a variable j has a domain of potential values D_j, the constraint on i and j, R_{ij}, is a subset of the Cartesian product of D_i and D_j. A pair of values (a, b) is called consistent, if (a, b) satisfies the constraint R_{ij} between i and j. The variables i and j are the *relevant* variables for R_{ij}. The constraint network is composed of the variables, the domains and the constraints. Thus, the problem is [9], [12] given a set of variables, a domain of possible values for each variable, and a conjunction of constraints, to find a consistent assignment of values to the variables so that all the constraints are satisfied simultaneously. CSP's are, in general, NP-complete problems and some are NP-hard [7]. Thus, a general algorithm designed to solve any CSP will necessarily require exponential time in problem size in the worst case.

3 NAIS: Network Artificial Immune System

We called our algoritm NAIS which stands for Network Artificial Immune System. The algorithm uses three immune components: antigen, antibody and B-cells. Basically, the antigen represents the information for each variable given by the constraint graph. This information is related to the number of connections of each variable, that is the number of constraints where each variable is a relevant one. Thus, it is a fixed information and not depends on the state of the search of the algorithm. On the contrary, the antibody strongly depends on the state of the search of the algorithm. It has two kinds of information: the variable values and the constraints violated under this instantiation. Finally, a B-cell has all the antibody information required by the algorithm to its evolution.

3.1 Immune Components for CSP

The immune components in our approach are defined as follows:

Definition 1. *(Antigen)*
For a CSP and its constraint graph we define the antigen Ag of the n-tuple of variables $(Ag_1, \ldots, Ag_n)$, such that the Ag_i value is the number of constraints where X_i is a relevant variable, $\forall i, i = 1, \ldots, n$.

The algorithm needs to know for each pre-solution its variable values and the constraints satisfied under this instantiation. For this reason, the antibody has two segments: a structural and a conflicting segment.

Definition 2. *(Structural antibody)*
A structural antibody Ab_s is a mapping from a n-tuple of variables $(X_1, \ldots, X_n) \rightarrow D_1 \times \ldots \times D_n$, such that it assigns a value from its domain to each variable in V.
Remark: The structural segment corresponds to an instantiation **I** *of the CSP.*

Definition 3. *(Conflicting antibody)*
For a CSP and its constraint graph we define the conflicting antibody Ab_c of the n-tuple of variables $(Ab_{c_1}, \ldots, Ab_{c_n})$, such that the Ab_{c_i} value is the number of violated constraints where X_i is a relevant variable, $\forall i, i = 1, \ldots, n$.

A solution consists of a structural antibody which does not violate any constraint, that is, whose conflicting antibody complements the antigen.

Before defining the B-cell we need to introduce the idea of affinity in the context of our problem.

3.2 Affinity Measure

In our approach we are interested in two kinds of affinity. The affinity between the antigen and a conflicting antibody, and the affinity between two structural antibodies.

- Affinity between the antigen and a conflicting antibody
  It is an estimation of how far an antibody is from being a CSP solution. It is related to the number of satisfied constraints by an antibody. The key idea is that a solution of the CSP corresponds to the biggest value of the affinity function between Ab_c and Ag. This occurs when all the constraints are satisfied. We define the function A_{csp} to measure this affinity as the euclidean distance computed by:

$$\mathbf{A_{csp}(Ag, Ab_c)} = \sqrt{\sum_{\mathbf{i=1}}^{\mathbf{n}} (\mathbf{Ag_i} - \mathbf{Ab_{c_i}})^{\mathbf{2}}} \tag{1}$$

  The function A_{csp} prefers a pre-solution with a minimal number of violated constraints as it is usual for guiding the search of incomplete techniques.
- Affinity between two structural antibodies
  Two structural antibodies has a high affinity level when they are quite similar in terms of the values of these variables. The idea of using this measure, named HA_s, is to quantify how similar two pre-solutions are. To compute this interaction our algorithm uses the Hamming distance. The algorithm uses this measure to obtain a diversity of pre-solutions.

3.3 B-Cell Representation

A B-cell is a structure with the following components:

- An Antibody $Ab = (Ab_c, Ab_s)$
- The number of clones of A_b to be generated for the clonal expansion procedure. This number is directly proportional to the A_{csp} value.
- The hypermutation ratio used in the affinity maturation step. This ratio is inversely proportional to the A_{csp} value.

3.4 The Algorithm - Network Artificial Immune System

The algorithm NAIS is shown in figure 1. NAIS works with a set of B-cells, following an iterative maturation process. Some of these B-cells are selected, doing a *clonal selection*, prefering those with bigger affinity values A_{csp}, that is, those that satisfy a greater number of constraints. It uses a Roulette Wheel selection, [17]. The algorithm generates a number of clones of the B-cells selected, that is done by the *clonal expansion* procedure. These clones follow a hypermutation process in the *affinity maturation* step.

The new set of B-cells is composed of a selected set of hypermutated B-cells.

```
Algorithm NAIS(CSP) returns memory B-cells
Begin
    $Ag \leftarrow$ Determine constraint graph connections($\mathbf{CSP}, n$);
    Initialize B-cells
    For i $\leftarrow$ 1 to $B - cells\_NUM$ do
        Compute affinity value $A_{csp}$(B-cells[i])
    End For
    j $\leftarrow$ 1;
    While (j $\leq$ $MAX\_ITER$) or (not solution) do
        Select a set of B-cells by Roulette Wheel
        Generate Clones of the selected B-cells
        Hypermutate Clones
        For k $\leftarrow$ 1 to $CLONES\_NUM$ do
            Compute affinity value $A_{csp}$(Clones[k])
        End For
        B-cells $\leftarrow$ build_network(CLONES);
        B-cells $\leftarrow$ metadynamics(B-cells);
    End While
    Return B-cells;
End
```

Fig. 1. NAIS Pseudocode

This selection is done in the *build_network* using the HA_s values in order to have a diversity of B-cells. A hypermutated B-cell could belong to the new set of B-cells if the hypermutated B-cell is quite different from the B-cells in memory. This difference is measured using the hamming distance, controlling the minimal degree of required diversity by the ϵ parameter value. Thus, a B-cell be accepted to be a new memory member when $(1 - \frac{HA_s}{n}) > \epsilon$. The ϵ value is known as the threshold of cross reactivity.

Finally, the algorithm adds new B-cells randomly generated in the *metadynamics* procedure to this set of B-cells. This procedure allows the algorithm to do more exploration of the search space.

Hypermutation procedure: The hypermutation is a hill-climbing procedure that repairs the conflicts in the conflicting antibody. This procedure is inspired on min-conflicts algorithm proposed in [16]. Figure 2 shows the hypermutation procedure.

```
Hypermutation(B-cell)
Begin
Repeat
$V$ = Select randomly a variable to be changed
If $Ab_c(V) > 0$ then
       Repeat
             Choose $v$ a new value for $V$ from its domain
             $NAb_c(V)$ = Number of conflicts for $V$ using $v$
             If $NAb_c(V) < Ab_c(V)$ then
               $Ab_s(V) = v$
               Re-compute $Ab_c$
             End If
       Until $(NAb_c(V) < Ab_c(V))$ or (Max_tries)
End If
Until Num_hyper
returns(B-cell)
End
```

Fig. 2. Hypermutation Procedure

Given a B-cell, a variable of its structural antibody is randomly selected to be changed. In case of the selected variable does not participate in any constraint violation (i.e. $Ab_c(V) = 0$), the procedure tries to modify another variable that could be involved in a conflict. The value of this variable is modified, such that, this value allows to reduce the number of conflicts recorded on the corresponding conflicting antibody. Thus, this procedure changes the structural antibody and also, as a consequence, the conflicting antibody.

The procedure Re-compute(Ab_c) does a partial evaluation of the conflicting antibody, only considering the variables related to the variable V. That is, just those variables which are relevants with V for a specific constraint. This kind of partial evaluation is quite useful in the constraint research community in order to reduce the computational time spent evaluating the constraints satisfaction for a new instantiation of the variables.

The hypermutation procedure uses two parameters: Max_tries and Num_iter. The parameter Max_tries corresponds to the maximum number of values to be tried for a given variable V. The parameter Num_iter corresponds to the maximum number of the B-cell variables that could be mutated.

4 Tests

The goal of the following benchmarks is to evaluate the performance of NAIS for solving CSP. The algorithm has been tested with randomly generated binary CSPs, [5]. The tests are to evaluate the NAIS behaviour when it is calibrated using the technique REVAC for tuning. We compare NAIS calibrated with GSA [4] that is a sophisticated evolutionary algorithm that solves CSPs and which strongly uses knowledge coming from the constraints research community. We also compare NAIS with SAW, [14]. SAW has been compared with both complete and incomplete well-known algorithms for CSP obtaining better results in most of the cases tested.

The hardware platform for the experiments was a PC Pentium IV Dual Core, 3.4Ghz with 512 MB RAM under the Mandriva 2006 operating system. The algorithm has been implemented in C. The code for NAIS is available in a web page.[1]

4.1 Problems Tested on the Hard Zone

The idea of these tests is to study the behavior of the algorithm solving hard problems. We use two models to generate binary CSPs. That is because GSA has been reported using model B proposed in [5] and SAW has been reported using model E, [14].

Model B: The binary CSPs belonging to the hard zone are randomly generated using the model proposed by B. Smith in [5]. This model considers four parameters to obtain a CSP. That is, the number of variables (n), the domain size for each variable (m), the probability that exists a constraint between two variables (p_1), and the probability of compatibility values(p_2). This model exactly determines the number of constraints and the number of consistent instantiations for the variables that are relevant for a given constraint. Thus, for each set of problems randomly generated the number of constraints are $\frac{p_1 n(n-1)}{2}$ and for a given constraint the number of consistent instantiations are $m^2 p_2$. Given (n, m, p_1) B. Smith defines a function to compute critical p_2 values, those values that allow to obtain CSPs on the transition phase, that is the problems that are harder to be solved.

$$\hat{\mathbf{p}}_{\mathbf{2}_{\text{crit}}}(\mathbf{n}, \mathbf{m}, \mathbf{p_1}) = \mathbf{m}^{-\frac{\mathbf{2}}{(\mathbf{n}-\mathbf{1})\mathbf{p_1}}} \tag{2}$$

Model E: This model also considers four parameters (n, m, p, k). The parameters n and m have the same interpretation than in model B. For binary CSPs whose constraints have two relevant variable $k = 2$ in model E. The higher p the more difficult, on average, problem instances will be.

[1] http://www-sop.inria.fr/orion/personnel/Marcos.Zuniga/CSPsolver.zip

4.2 REVAC

The Relevance Estimation and Value Calibration has recently been proposed in [15]. The goal of this algorithm is to determine the parameter values for evolutionary algorithms. It is also able to identify which operators are not relevant for the algorithm. Roughly speaking, REVAC is a steady-state evolutionary algorithm that uses a real-value representation, where each value corresponds to a parameter value of the algorithm to be tunned. Each chromosome in REVAC is evaluated by the performance obtained by the algorithm (to be tunned) using its parameter values. A new individual is randomly created, however the value for each variable is obtained only considering the values that are in the population for this variable. It does 1000 evaluations.

In order to apply REVAC for calibrating NAIS, we have selected 14 problems, two from each category $< 10, 10, p_1, p_2 >$ using model B. The performance for each chromosome is computed as the number of satisfied constraints by the solution obtained by NAIS using these parameter values. The parameter values found by this tuning procedure were:

- $n_1 = 0.3$, rate of cells to be expanded,
- $n_2 = 0.4$, rate of cells to be incorporated on the memory
- $\epsilon = 0.40$, threshold reactivity between clones
- $B - cells = 5$
- Number of clones = 100

That means that NAIS requires to do more exploration than it does using a hand-made calibration. In the hand-made calibration the hypermutated cell is accepted if it differs at least in a 54% ($\epsilon = 0.46$) from the memory cells. Now, it must differ at least in a 60% to be accepted. Furthermore, the number of cells to be expanded has been reduced in 0.2. The procedure required around 14 hours, computational time, to determine these parameter values.

4.3 Tests with Calibration

Comparison between NAIS and GSA using Model B: Because the reported results of GSA, [4] has been evaluated with the problems in the hardest zone as they have been generated in [5], we run the calibrated NAIS using

Category	GSA		NAIS					
	50.000 ev.	time [s]	10.000 ev.	time [s]	50.000 ev.	time [s]	75.000 ev.	time [s]
c0.3_t0.7	93.33	2.18	87.5	0.68	93.75	1.85	97.5	1.98
c0.5_t0.5	84.4	2.82	91	0.61	98.33	1.08	99.33	0.98
c0.5_t0.7	100	2.76	84	0.61	83.33	2.79	84.33	3.91
c0.7_t0.5	16.7	10.35	87	0.77	87	2.09	90.67	3.34
c0.7_t0.7	100	2.51	80.67	0.66	80.33	2.15	82	4.52
c0.9_t0.5	3.3	13.77	83.67	0.52	86.33	2.15	87.33	4.19
c0.9_t0.7	99.0	1.58	75.33	0.82	75.33	3.92	75.67	6.03

Fig. 3. Success rate and CPU time for NAIS and GSA

problems generated using the parameters (n, m, p_1, p_2). We consider the problems with $n = m$, where $n = 10$. The p_1 and p_2 values are those belonging to the hardest zone. In figure 3 *c*0.3*_t*0.7 means $p_1 = 0.3$ and $p_2 = 0.7$. The following table shows the percentage of problems solved and the time required for GSA and those obtained by NAIS considering 10.000, 50.000 and 75.000 evaluations.

NAIS has a higher satisfaction rate than GSA, moreover it converges very quickly to good solutions. Furthermore, considering just 10.000 evaluations the average succes rate for NAIS was around 83% instead of 72% for GSA. However, in some categories GSA outperforms NAIS.

p	SAW		NAIS					
	100000 ev.		10000 it.		50000 it.		75000 it.	
	succes	time [s]	succes	time [s]	succes	time [s]	succes	time [s]
0.24	100	0.74	100	0.32	100	0.33	100	0.3
0.25	100	2.33	100	0.44	100	0.43	100	0.4
0.26	97	6.83	100	0.56	100	0.6	100	0.61
0.27	60	11.39	100	1.2	100	1.1	100	1
0.28	25	18.66	98.4	2.06	100	2.26	100	2.05
0.29	17	20.57	84	4.11	99.6	5.24	100	5.41
0.3	5	22.27	47.6	6.99	84.4	17.17	90	21.02
0.31	1	22.47	16.8	8.62	38.4	35.1	46.8	48.91
0.32	0	22.39	24	8.22	59.6	27.64	63.6	37.95
0.33	1	22.38	24.4	8.3	59.6	27.4	68.4	34.85

Fig. 4. Success rate and CPU time for NAIS and SAW

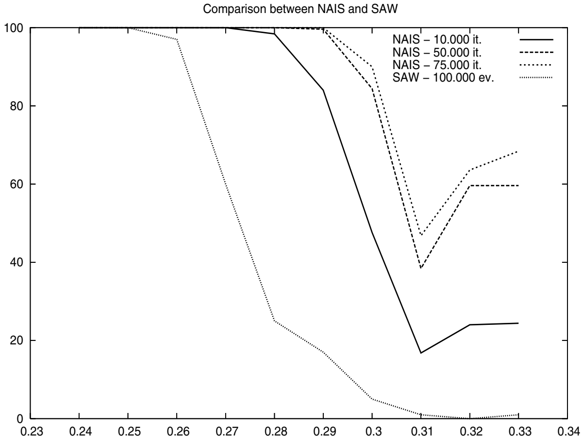


Fig. 5. Different problems tested, comparison of % Succesful runs

Comparison between NAIS and SAW using Model E: We have generated 250 problem instances in the hard zone using Model E. Figure 4 shows the success rate and the time in seconds for both algorithms.

We can observe that NAIS outperforms SAW in both time and success rate. Moreover, the average success rate for SAW is 40.6% instead of a 69.2% for NAIS, just considering 10.000 iterations. NAIS required, for these number of iterations, in average, just 4.1 seconds.

Figure 5 shows the results for NAIS and SAW. We can observe in NAIS the transition phase in $p = 0.31$.

5 Conclusions

The artificial immune systems have some interesting characteristics from the computational point of view: pattern recognition, affinity evaluation, immune networks and diversity. All of these characteristics have been included in our algorithm. The B-cell structure is useful to determine both the solution of the problems and also to identify conflicts. The conflicting antibody is used by the algorithm to guide the reparation of the solutions (hypermutation process), giving more priority to the variables involved in a higher number of conflicts. For the problems in the hardest zone NAIS just using 10.000 iterations (avg. 4.1 seconds) solved, on average, 28% more problems than SAW, one of the best known evolutionary algorithm. The calibrated NAIS solved more problems than GSA, that is a sophisticated genetic algorithm which incorporated many constraints concepts to solve CSP. Artificial Immune Systems is a promising technique to solve constrained combinatorial problems.

6 Future Work

A promising research area is to incorporate some parameter control strategies into the algorithm. The tuning process to define the parameter values for NAIS has been a time consuming task.

References

[1] de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, London (2002)

[2] de Castro, L.N., von Zuben, F.J.: Learning and Optimization Using the Clonal Selection Principle. IEEE Transactions On Evolutionary Computing 6(3), 239–251 (2002)

[3] Dasgupta, D.: Artificial Immune Systems And Their Applications. Springer, Heidelberg (2000)

[4] Dozier, G., Bowen, J., Homaifar, A.: Solving Constraint Satisfaction Problems Using Hybrid Evolutionary Search. IEEE Transactions on Evolutionary Computation 2(1), 23–33 (1998)

[5] Smith, B., Dyer, M.E.: Locating the phase transition in constraint satisfaction problems. Artificial Intelligence 81, 155–181 (1996)

[6] Timmis, J., Neal, M.: Investigating the Evolution and Stability of a Resource Limited Artificial Immune System. In: Proceedings of the IEEE Brazilian Symposium on Artificial Neural Networks, pp. 84–89 (2000)

[7] Cheeseman, P., Kanefsky, B., Taylor, W.: Where the Really Hard Problems Are. In: Proceedings of IJCAI-91, pp. 163–169 (1991)

[8] Eiben, A.E., van Hemert, J.I., Marchiori, E., Steenbeek, A.G.: Solving Binary Constraint Satisfaction Problems using Evolutionary Algorithms with an Adaptive Fitness Function. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) Parallel Problem Solving from Nature - PPSN V. LNCS, vol. 1498, pp. 196–205. Springer, Heidelberg (1998)

[9] Mackworth, A.K.: Consistency in network of relations. Artificial Intelligence 8, 99–118 (1977)

[10] Marchiori, E.: Combining Constraint Processing and Genetic Algorithms for Constraint Satisfaction Problems. In: Proceedings of the 7th International Conference on Genetic Algorithms (ICGA97) (1997)

[11] Riff, M.-C.: A network-based adaptive evolutionary algorithm for CSP. In: The book Metaheuristics: Advances and Trends in Local Search Paradigms for Optimisation, ch. 22, pp. 325–339. Kluwer Academic Publisher, Boston, MA (1998)

[12] Tsang, E.P.K., Wang, C.J., Davenport, A., Voudouris, C., Lau, T.L.: A family of stochastic methods for constraint satisfaction and optimization. In: Proceedings of the 1st International Conference on The Practical Application of Constraint Technologies and Logic Programming (PACLP), London, pp. 359–383 (1999)

[13] Solnon, C.: Ants can solve Constraint Satisfaction Problems. IEEE Transactions on Evolutionary Computation 6(4), 347–357 (2002)

[14] Craenen, B., Eiben, A.E., van Hemert, J.I.: Comparing Evolutionary Algorithms on Binary Constraint Satisfaction Problems. IEEE Transactions on Evolutionary Computation 7(5), 424–444 (2003)

[15] Nannen, V., Eiben, A.E.: Relevance Estimation and Value Calibration of Evolutionary Algorithm Parameters. In: Proceedings of the Joint International Conference for Artificial Intelligence (IJCAI), pp. 975–980 (2007)

[16] Minton, S., Johnston, M., Philips, A., Laird, P.: Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems. Artificial Intelligence 58, 161–205 (1992)

[17] Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer, Heidelberg (1992)

[18] Garret, S.: Parameter-free adaptive clonal selection. IEEE Congress on Evolutionary Computation (2004)

A Solution Concept for Artificial Immune Networks: A Coevolutionary Perspective

Oscar Alonso, Fabio A. Gonzalez, Fernando Niño, and Juan Galeano

Intelligent Systems Research Lab - National University of Colombia
{omalonsom,fagonzalezo,lfninov,jcgaleanoh}@unal.edu.co

Abstract. In this paper a relation between artificial immune network algorithms and coevolutionary algorithms is established. Such relation shows that these kind of algorithms present several similarities, but also remarks features which are unique from artificial immune networks. The main contribution of this paper is to use such relation to apply a formalism from coevolutionary algorithms called *solution concept* to artificial immune networks. Preliminary experiments performed using the aiNet algorithm over three datasets showed that the proposed solution concept is useful to monitor algorithm progress and to devise stopping criteria.

1 Introduction

Several artificial immune network (AIN) algorithms have been developed based on Jerne's idiotypic network theory. They have been applied with success to several problems, including clustering, classification, optimization, and domain specific problems [3]. These algorithms present several similarities with evolutionary algorithms: an AIN is a population based meta-heuristic which uses variation and selection mechanisms over a population of individuals. Moreover, an AIN has elements in common with coevolutionary algorithms, a branch of evolutionary algorithms in which individuals interact between them. In an AIN, the stimulation of a cell is influenced not only by the antigens it detects, but also by other cells of the AIN.

Therefore, analyzing the relation between AINs and coevolutionary algorithms may provide interesting insights to each domain. Coevolution has been subject of theoretical analysis [4], which could provide AINs with effective stopping criteria, performance metrics, and a strong theoretical background. Also, coevolutionary algorithms may be improved by taking elements from AINs, such as diversity introduction (metadynamics), memory mechanisms, and dynamic regulation of population size. Some works have already successfully explored the relation between such models[10].

In this work AIN algorithms are analyzed from a coevolutionary point of view, by identifying common elements in both models, and also examining the differences between AIN algorithms and coevolutionary algorithms. Later, the *solution concept* formalism from coevolutionary algorithms is applied to AIN models, in order to characterize the expected result of the AIN dynamics. In addition, preliminary experimentation is performed on an existing AIN algorithm

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 35–46, 2007.

(aiNet), in order to test the potential use of the proposed solution concept to define stopping criteria and performance metrics.

The rest of the paper is organized as follows. Section 2 presents a background of coevolutionary algorithms. Section 3 analyzes artificial immune networks from a coevolutionary point of view, showing similarities and differences between such two models. In section 4, the solution concept formalism from coevolutionary algorithms is applied to AINs, and a solution concept is defined for AINs. Results of exploratory experimentation are shown in section 5, which suggests the potential use of the proposed solution concept to monitor the behavior of AIN algorithms . Finally, some conclusions are devised in section 6.

2 Coevolutionary Algorithms

An evolutionary process leads to an increase on the average fitness of a population. The main mechanisms of evolution are variation and selection; selection is performed according to a fitness measure, favoring better fit individuals, i.e., fittest individuals have a higher probability to survive.

Coevolution occurs when a change in individuals from one species induces evolution on individuals of another species. For example, suppose that a plant is subject to attack by an insect. The plant population may undergo evolution to develop toxins to defend itself from insects. Then, such toxins can induce evolution on the population of insects, for them to develop resistance to the toxin. Thus, the evolution of the plants induces a selection pressure over the insects population, which results in the evolution of the population of bugs.

From a computational viewpoint, a coevolutionary algorithm is characterized by a fitness function that changes as the population evolves, and is usually defined based on interaction with other individuals whom are also evolving.

2.1 One vs Many Populations

Coevolutionary algorithms usually involve more than one population of individuals. Each population represents a species, and the individuals of one population are evaluated according to their interaction with individuals of other populations.

It is also considered that coevolution arises when an individual is evaluated according to its interaction with individuals of its own population.

2.2 Competitive vs Cooperative Coevolution

The interaction between individuals can have different purposes, which can be mainly classified in two kinds of interaction: competitive and cooperative.

In competitive coevolution, each individual represents one solution to a problem, and the interaction between individuals is used to test individuals' features: the solution obtained is said to be test-based. An example of this interaction is when players for games are evolved: individuals receive fitness according to the results of their games against other individuals.

In cooperative coevolution, the output of the algorithm is composed of several individuals, which together represent a solution to the problem. Individuals interact in order to measure not only their features, but also their synergy. Here, the solution is said to be compositional. A very simple example of this approach is the optimization of a two variables function *f(x,y)*: the solution can be decomposed into two elements (x and y coordinates) and each value is evolved in its own population. Individuals are evaluated by selecting a random mate from the other population and evaluating the target function *f(x,y)*.

It is important to notice that in some cases the interaction presents characteristics from both categories: then, the interaction is said to be mixed.

2.3 Interaction Between Individuals

In most coevolutionary algorithms the interaction between individuals is direct: individuals meet each other and obtain fitness from such encounter. But, in some coevolutionary models, the interaction between individuals is not direct: their interaction occurs through their relation with a common environment, for instance, by consuming resources in a common environment. An example is the Lotka-Volterra model, in which predators interact directly with preys (by eating them), but they also interact by competing for a scarce resource: when a predator eats a pray, the remaining amount of prays for other predators diminishes.

Some well-known techniques in evolutionary algorithms which involve indirect interaction can be considered coevolutionary. An example is a niching technique known as *competitive fitness sharing*[4], used in multi-objective optimization. In this technique, the fitness produced by satisfying an objective is distributed among the individuals that are able to fulfill it. Thus, individuals that satisfy objectives that others do not are rewarded, promoting diversity in the population.

2.4 Generational vs Steady State Algorithms

In generational evolutionary algorithms, all individuals of the population are replaced by their descendants. In contrast, in steady state evolutionary algorithms, some individuals are added to the population and then some are removed, allowing an individual to remain in the population for several generations.

2.5 Spatially Embedded Algorithms

Usually, the interaction between individuals happens by choosing other individuals at random from their respective population. However, in some cases, individuals are embedded in a space, and the interaction only occurs between individuals located in a neighborhood. Frequently, such space corresponds to a lattice, and a local rule only allows individuals to interact with a limited amount of their neighbors [8].

The interaction space does not need to be explicit: it can be implicitly defined by a metric that represents a similarity measure between individuals. For instance, a niching technique known as *similarity based fitness sharing* [4] involves interaction between individuals of the population, and, therefore, it can be considered coevolution. In this technique, the fitness of an individual decreases as the number of similar individuals in the population increases. It should be emphasized that in this case individuals interact according to locality in the genotype space.

In general, spatial embedding has been used to improve diversity in coevolutionary algorithms. Thus, spatiality does not represent any feature in the problem domain, but it is simply a convenient tool to improve an algorithm's performance[11].

2.6 Solution Concepts

A solution concept is a set of criteria used to determine which elements of a search space can be considered as solutions to a particular search problem. A solution concept partitions the search space into solution regions and non-solution regions. Therefore, a population-based algorithm should be designed in such a way that the population converges to solution regions. It is important to notice that a solution concept is specific to a particular search problem.

Every search problem defines a solution concept, which characterizes what is being searched. Also, each search algorithm implements a solution concept, which corresponds to what it actually searches. Sometimes, coevolution dynamics generates unexpected behavior. According to Ficici [4], this is related to a mismatch between the intended solution concept and the algorithm's solution concept. Therefore, solution concepts are a formalism useful in the design of proper algorithm dynamics, performance measures, and stopping criteria.

3 Immune Networks and Coevolution

Idiotypic network theory (also known as immune network theory) was developed by Jerne [6,9] in order to explain the memory and distributed self regulated behavior of the immune system. Such theory states that immune system cells are able to react not only to pathogens, but also to other elements of the immune system. During antigen exposure, clones (groups of identical cells) which recognize the antigen increase their size. When the antigen disappears, most of those cells would die and the clones would shrink or even disappear if not stimulated. Immune network theory states that a B cell can be stimulated by other cells of the immune system, and that this feedback mechanism can maintain the cells that were created during exposure to an antigen.

3.1 Artificial Immune Networks

An artificial immune network (AIN) is a computational model built upon Jerne's idiotypic network theory. It is a population based meta-heuristic, which develops a set of detectors (B cells[1]) that interact with data (antigens) and with each

[1] As each B cell express only one kind of antibody, some models consider antibodies directly, instead of B cells.

other. AINs perform unsupervised learning; they have been typically used for clustering, but have also been adapted to optimization, classification and domain specific applications [5].

AIN dynamics include interaction with antigens and between detectors. As a result of those interactions detectors become stimulated. Highly stimulated B cells undergo cloning and somatic hypermutation. Somatic hypermutation is a genetic operator in which the mutation rate depends on the affinity of the cell with the current stimulating antigen. This process is regulated by the interaction between B cells, which can stimulate them in order to create a memory of observed antigens, or suppress them, in order to control the immune response.

AIN algorithms also posses a *metadynamic*s, which includes natural death of unstimulated detectors and addition of new random detectors to the population.

3.2 A Comparison Between AINs and Coevolution

Here, similarities between AIN and coevolutionary algorithms are presented by elicitating a correspondence between their elements, which is summarized in Table 1.

In an AIN, the population of individuals is the set of B cells, and every B cell is usually represented as a real valued vector or a binary string. These individuals are stimulated by antigens and by other individuals, and the resulting stimulation level correspond to the subjective fitness of a coevolutionary algorithm.

Table 1. Parallel between coevolutionary algorithms and artificial immune networks. Notice that there is no equivalence in coevolutionary algorithms for the antigens and for the addition of new random cells to the population.

Coevolution	**Immune networks**
Individual	B cell
Population	Set of B cells (variable size)
-	Antigens
Subjective fitness	B cell stimulation level
Selection	Selection for cloning according to stimulation Natural death (metadynamics)
Reproduction	Cloning
Genetic operators	Somatic hypermutation
Competitive interaction	Suppression between B cells Competition for resources (in models with limited resources)
Cooperative interaction	The output of the algorithm is a set of detectors that cooperatively solve a problem Coestimulation between B cells
Spatiality	There is a notion of space (shape space), defined by the affinity measure. Detectors interact directly only with elements on their neighborhood in the shape space
-	B cell born (metadynamics)
Solution concept	Depends on the problem to be solved Not explicitly specified

An AIN's dynamics is similar to a steady state genetic algorithm, as a B cell can survive through several generations. A B cell may be selected for removal in metadynamics if it has a very low level of stimulation. Also, if it has a high level of stimulation, it undergoes cloning (asexual reproduction), and descendant cells suffer a mutation inversely related to the affinity with the current antigen (somatic hypermutation).

An AIN presents a mixed interaction between individuals; first, B cells interact in a cooperative way to solve a problem, that is, the result of the algorithm is the whole set of cells, rather than a single cell. Also, in models that consider limited resources, the B cells compete between them for such resources. Additionally, B cells interact between them to suppress or stimulate each other, which can be considered cooperative and competitive behavior respectively. Artificial models usually define a stimulation function and a suppression function, both of which depend on the affinity of a detector with its neighbors.

Individuals interact with each other and with antigens according to their affinity, which correspond to their similarity (in most cases) or complementarity. In other words, they interact with each other if they are neighbors in the shape space[2]. For real valued vectors the affinity metric is typically the Euclidean distance or a function of it . In the case of strings the Hamming distance, the edit distance, or a function of them could be used.

3.3 Differences Between AINs and Coevolution

Although AINs present several similarities with coevolutionary algorithms, they also present some particularities that make them different from coevolutionary algorithms.

In AINs, interaction is defined spatially, not as a way to improve an algorithm performance or to promote diversity, but because the model is inherently spatial. Also, a memory mechanism is inherent to immune networks as a result of its dynamics, while several coevolutionary algorithms implement some kind of memory in order to avoid cyclic dynamics, but as a external mechanism.

Additionally, immune network models are self regulatory, which means that they are free to vary population size, but this has to be done in a reasonable way, since there has to be an equilibrium state instead of an explosion in the number of cells.

Regarding solution concepts, there is no clear solution concept related to immune networks. Though the purpose of the algorithm is usually described, the properties a solution should have are rarely specified. Given the output of an AIN algorithm, clustering of the data is usually accomplished by connectivity of the detectors [7](each connected component of the network is a cluster), or by applying MST algorithm to the detector set[2].

In the next section, a solution concept for artificial immune network algorithms is proposed.

[2] A physical space in which B cells move (such as the blood torrent) is never considered.

4 A Solution Concept for Artificial Immune Networks

As discussed above, the solution concept formalism may be useful for characterizing AINs. Specifically, it could help to define performance measures, devise stopping criteria, and to refine algorithm dynamics in order to reach a desired solution. Therefore, here the purpose of the immune network model, its input and its output are characterized.

4.1 Characterization of the Solution Concept

Although AIN models have been adapted to perform a variety of tasks, here the purpose of AINs will be restricted and stated in machine learning terminology, in order to provide a precise definition of the algorithm and a unified framework to compare different algorithms. This solution concept is based on some concepts, such as probability density estimation, which will be specified later. An AIN algorithm is restricted to fit these criteria:

- An AIN algorithm performs unsupervised learning: The algorithm does not receive information about the labels of incoming data.
- An AIN algorithm performs semi-parametric probability density estimation: The output of the algorithm provides a model of the data, by representing the probability distribution of the training data.

Therefore, for a set of antibodies produced by an algorithm to be considered a solution, it must satisfy the following criteria:

- Equilibrium: An AIN is in equilibrium if it keeps its structure without change (or with a small change) indefinitely (or for a given period of time), given that there is not external stimuli. This criterion is related to the AIN memory mechanism.
- High antigen set modeling capability: AIN's cells should represent the distribution of antigens in the shape space. In other words, the probability distribution that it estimates should be as close as possible to the one that generated the antigens.

Therefore, the solution concept is defined as a subset of equilibrium states which satisfy these criteria.

Next, a characterization of AINs as probability density estimators is presented, as well as the construction of such probability density function, and measures associated to the proposed solution concept.

4.2 AINs as Semi-parametric Probability Density Estimators

Probability density estimation methods try to estimate a probability distribution P based on a set of samples of it. These methods can be classified into three categories: parametric, non-parametric and semi-parametric methods [1].

In parametric learning the data is supposed to be generated from some probability distribution, and the learning algorithm tries to find suitable values for the parameters of such distribution in order to fit the data.

In contrast, in non-parametric models an a priori distribution is not assumed. The model is built by associating a kernel function (usually Gaussian) to each point of the training data. Then, the model is constructed as a sum of such kernels. These methods rely on the fact that any probability distribution can be approximated as a mixture of Gaussians (see Fig.1). However, they require to store all the training data, and their use is computationally demanding.

In semi-parametric methods a set of kernels is obtained, but they do not correspond to the whole set of the training data. Only a few kernels are generated, which represent a model of the data. In this case as in the parametric case, parameters of the kernels are selected in order to fit the data in a better way.

AIN algorithms aim to obtain a set of B cells located in regions with high concentration of antigens. Then, if each B cell is assigned a Gaussian kernel, the generated probability distribution mimics the probability density of appearance of new antigens in a given region. Therefore, AINs algorithms can be considered semi-parametric density estimation methods, which try to model the probability density distribution of the antigens in the shape space, using a few number of kernels, each one associated to a B cell.

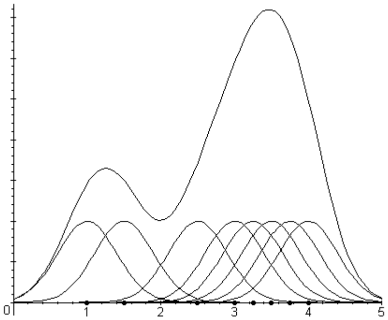


Fig. 1. Mixture of Gaussian kernels: The set of kernels generate the probability function shown. Notice that all Gaussian kernels have the same standard deviation and the same weight. However, in general, they can have different values for the standard deviation, and each kernel can have a different weight.

4.3 Antibody Set as a Model of Input Data

Input data will be assumed to be a set (training set TS) of n-dimensional real valued vectors (ag_i, $i \in \{1...N\}$), which are assumed to come from an unknown probability distribution P. The data is unlabeled, and may include noise.

After antigen exposure, the population of antibodies reaches a configuration (not necessarily static), which is the output of the algorithm. Such output is a set of detectors (antibodies ab_i, $i \in \{1...M\}$), along with the number of cells or concentration of each clone (c_i, $i \in \{1...M\}$), for the algorithms that use the ARB representations), or the stimulation, for algorithms that consider individual antibodies. The detectors are also n-dimensional real valued vectors.

An antibody set is interpreted as a probability distribution in this way:

- Each detector ab_i is the center of a Gaussian kernel $k_i(x) \sim N(\mu_i, \sigma^2)$, where $\mu_i = ab_i$
- The standard deviation of the Gaussian kernel is the radius associated with the detector.
- The weight of each kernel is either the concentration (number of resources, for ARBs) or the stimulation level (for models representing individual B cells)

Accordingly, an antibody set is then interpreted as a probability distribution as follows:

$$\hat{P}(x) = (\textstyle\sum_{i=1}^{M} k_i(x)c_i)/C, \text{ where } C = (\textstyle\sum_{i=1}^{M} c_i)$$

4.4 AIN Modeling Capability

A measure used to calculate how well a probability distribution fits the data is the Likelihood. The likelihood expresses the probability that a dataset had been generated by a given probability distribution.

Therefore, the likelihood of $\hat{P}$ over the antigens will be used as a measure of how well a set of antibodies models the antigen set. For numerical reasons, it is better to consider the log-likelihood, which is defined as

$$L = \textstyle\sum_{j=1}^{N} ln(\hat{P}(ag_j)) = \sum_{j=1}^{N} ln(\sum_{i=1}^{M} k_i(ag_j)c_i)$$

A high likelihood implies that the difference between the distributions P (the probability distribution from which antigens are sampled) and $\hat{P}$ (the estimated probability distribution) is expected to be minimum. Therefore, an AIN will be required to present a high likelihood over the antigens in order to be considered a solution.

4.5 Equilibrium

Memory requires that once a pattern is found in the data, it will not vanish later. Therefore, equilibrium is a requirement for memory.

Equilibrium implies that the likelihood of the network over the training data and the number of detectors do not vary significantly through time.

Thus, equilibrium can be tested by letting the AIN iterate without antigen presentation, and monitoring the number of network's cells and its likelihood over the data.

5 Exploratory Experimentation

Some preliminary experimentation was carried out in order to evaluate the antigen set modeling capability requirement of the solution concept.

The aiNet model, proposed by de Castro [2], was used to measure likelihood of the network using two synthetic datasets (the two spiral dataset and a two cluster dataset), and the IRIS dataset. Results are shown in figures 2, 3 and 4, respectively. In all cases, the parameters of the algorithm were set as follows: suppression threshold was set to 0.7, n=4, N=10, and qi=0.1. For the probability estimation, the standard deviation was taken as the suppression threshold, and all the kernels were assigned the same weight.

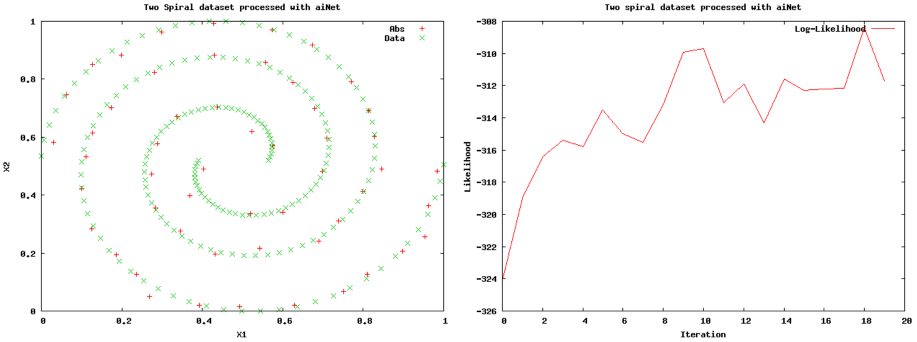


Fig. 2. Output and Likelihood evolution of the aiNet algorithm over the two spirals dataset (190 items)

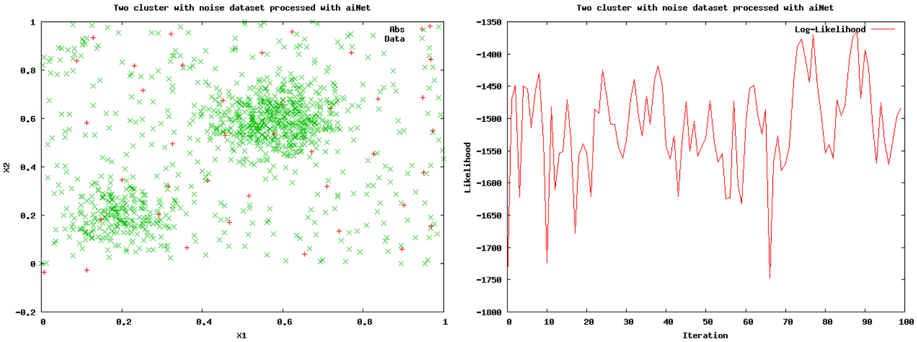


Fig. 3. Output and Likelihood evolution of the aiNet algorithm over the two clusters dataset (1018 items)

Results for the two spiral and the IRIS dataset show how the likelihood of the network increases along iterations, which indicates that it is a good measure of the algorithm progress. Also, results from the IRIS dataset show that the likelihood reaches a stable value about iteration 20, and later there is no further improvement in the likelihood. This suggest the use of the likelihood as a measure of convergence, and, therefore, as a stopping criterion: a stable value after several iterations suggest that the algorithm has converged.

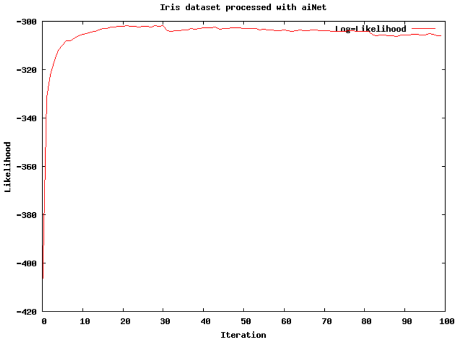


Fig. 4. Likelihood evolution of the aiNet algorithm over the IRIS dataset (150 items)

In the case of the two cluster dataset, the resulting configuration of the network does not resemble the distribution of input data. Therefore, the algorithm has failed to find an appropriate model of the data. It should be mentioned that the parameters of the AIN were set to provide such output. It is important to notice that the likelihood of the network does not increase throughout time, instead it shows an erratic behavior. Such behavior is an indicator of problems in the learning process, and suggest that the likelihood can be used to monitor the algorithm's performance.

6 Conclusions

In this paper, a formal relation between coevolutionary algorithms and artificial immune networks was established. It was shown that coevolutionary algorithms and AINs have many elements in common, but also that AINs present some characteristics that are not present in coevolutionary algorithms, such as variable size population, self regulatory behavior, inherent memory and spatiality.

Following Ficici's observations about coevolutionary dynamics, a solution concept for AINs was defined. Particularly, AINs are considered as a semi-parametric technique for probability density estimation: they estimate the probability function from which antigens are assumed to be sampled.

The proposed solution concept required the probability function estimated by the algorithm to have optimal likelihood over the training data; it also expects that the structure of the network remain stable in absence of antigens.

Network's likelihood was tested in the aiNet algorithm, and preliminary experimental results showed that it increased as the learning process took place, suggesting its use to monitor the algorithm's progress. Additionally, highly oscillatory likelihood over time was shown to indicate problems in the learning process.

A solution concept is useful to analyze the behavior of an AIN algorithm: results showed that it can be used to monitor algorithm's performance, and accordingly, to formulate stopping criteria. In addition, a solution concept may help to develop new algorithms or to improve existing ones.

As semi-parametric probability estimators, AINs have the advantage that they implicitly find the suitable number of required kernels.

Future work will focus on analyzing convergence properties of current AIN algorithms based on the proposed solution concept. Also, analytic work will be undertaken in order to relate immune network dynamics to the regularization theory, which attempts to limit the complexity of machine learning algorithms in order to achieve better generalization.

References

1. Bishop, C.M.: Neural networks for pattern recognition. Oxford University Press, Oxford, UK (1996)
2. de Castro, L.N., Von Zuben, F.J.: An evolutionary immune network for data clustering. In: França, F.M.G., Ribeiro, C.H.C. (eds.) SBRN, pp. 84–89. IEEE Computer Society, Los Alamitos (2000)
3. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Approach. Springer, London. UK (2002)
4. Ficici, S.G.: Solution Concepts in Coevolutionary Algorithms. PhD thesis, Computer Science Department. Brandeis University, USA (May 2004)
5. Galeano, J.C., Veloza-Suan, A., González, F.A.: A comparative analysis of artificial immune network models. In: GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation, pp. 361–368. ACM Press, New York, NY, USA (2005)
6. Jerne, N.K.: Towards a network theory of the immune system. Annals of Immunology 125C, 373–389 (1974)
7. Knight, T., Timmis, J.: Aine: An immunological approach to data mining. In: Cercone, N., Lin, T., Wu, X. (eds.) IEEE International Conference on Data Mining, pp. 297–304. IEEE, San Jose, CA. USA (2001)
8. Mitchell, M., Thomure, M.D., Williams, N.L.: The role of space in the success of coevolutionary learning. In: Artificial Life X: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems. International Society for Artificial Life, pp. 118–124. The MIT Press(Bradford Books), Cambridge (2006)
9. Perelson, A.S., Weisbuch, G.: Immunology for physicists. Reviews of Modern Physics 69, 1219 (1997)
10. Potter, M.A., De Jong, K.A.: The coevolution of antibodies for concept learning. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) Parallel Problem Solving from Nature - PPSN V. LNCS, vol. 1498, pp. 530–539. Springer, Heidelberg (1998)
11. Wiegand, R.P., Sarma, J.: Spatial embedding and loss of gradient in cooperative coevolutionary algorithms. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) Parallel Problem Solving from Nature - PPSN VIII. LNCS, vol. 3242, pp. 912–921. Springer, Heidelberg (2004)

Artificial Immune Systems for Classification of Petroleum Well Drilling Operations

Adriane B.S. Serapião[1], José R.P. Mendes[2], and Kazuo Miura[3]

[1] UNESP/IGCE/DEMAC – C.P. 178 – CEP 13506-900 – Rio Claro (SP), Brazil
adriane@rc.unesp.br
[2] UNICAMP/FEM/DEP – C.P. 6122 – CEP 13081-970 – Campinas (SP), Brazil
jricardo@dep.fem.unicamp.br
[3] PETROBRAS, E&P-SSE/UN-BC/ST/EP – Macaé (RJ), Brazil
kazuo.miura@petrobras.com.br

Abstract. This paper presents two approaches of Artificial Immune System for Pattern Recognition (CLONALG and Parallel AIRS2) to classify automatically the well drilling operation stages. The classification is carried out through the analysis of some mud-logging parameters. In order to validate the performance of AIS techniques, the results were compared with others classification methods: neural network, support vector machine and lazy learning.

Keywords: Petroleum Engineering, mud-logging, artificial immune system, classification task.

1 Introduction

Offshore petroleum well drilling is an expensive, complex and time-consuming operation and it demands a high qualification level from the drilling executors. One of the trends of the oil industry is the application of real-time measurements and optimization of production operations with the purpose of guaranteeing a safe and effective/low cost drilling execution. The concept of *digital fields* has been widely used in current works to denote continuous optimization of production [1]. This trend has also been seen in drilling, as real-time measurements and control are as well gaining attention in this particular area. In the last two decades, the technological advances in drilling techniques have notably contributed to the lowering of costs and to the expansion of exploration areas.

Technological progress in the petroleum engineering area was partly motivated by the evolution in instrumentation techniques, which affected not only the exploration segment but also the production one. As a result of the increasing instrumentation level, today, there is a lot of data being measured and recorded. But the techniques of data interpretation and evaluation have not developed at the same speed, and there is a lack of tools able to make an efficient use of all the data and information available.

This work presents the development of a system that intends to make better use of the information collected by mud-logging techniques during well drilling operations. The mud-logging techniques collected a great amount of data, and these data nowadays, are being used in a superficial way. The proposed system aims to take advantage of some of the information potential that is still not being used.

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 47–58, 2007.

The proposed methodology is able to generate a precise report of the execution stages during an operation through the interpretation of mud-logging data. There are two possible applications. The first one is related to the performance analysis and normality investigations. In this sense, this tool could be used to carry out the latter analysis of the time spent drilling each well in a field and to investigate how much of the total operation time each stage consumed and based on this statistics to plan the drilling of other wells. The second one is related to the production of an on-line logging of the executed stages. The methodology could be used on-line in the rig, so the system would be able to produce a report of the execution stages, and this report will present the same time precision as that of the mud-logging data.

The main idea of this work is that there is a great amount of information that has not been properly used, and this information could be used to provide a process feedback and to produce performance enhancements. There are initiatives of development of automatic monitoring systems in other areas like the work presented by Yue *et al.* [2] in Mining Engineering.

Information concerning individual drilling performance can also be used to build benchmarking analysis. In this sense, a petrol company could use this information to compare the performance of different divisions. On a minor scale, the company could compare performance of rented rigs and identify weak points as part of ongoing improvement process. The results produced by an automatic classification system may help in the design of new wells. The information about the time spent to execute a determined stage could be used for planning new wells in the same region providing cost estimates.

Artificial Immune Systems (AIS) are a new class of algorithms inspired by how the immune system recognizes attacks and remembers intruders [3]. The recognition and learning capabilities of the natural immune system have been an inspiration for researchers developing algorithms for a wide range of application. In this paper we are interested in applicability of artificial immune systems for real world data mining, and classification is one of the most important mining tasks, so we focus on the Clonal Selection Algorithm (CLONALG) and on the Artificial Immune Recognition System (AIRS) algorithm for that task.

CLONALG was proposed in 2000 [4] and is based on the clonal selection principle, which is used by the immune system to describe the basic features of an immune response to an antigenic stimulus. It establishes the idea that only those cells that recognize the antigens proliferate, thus being selected against those that do not. The selected cells are subject to an affinity maturation process, which improves their affinity to the selective antigens. The computational implementation of the clonal selection algorithm takes into account the affinity maturation of the immune response [5].

AIRS was introduced by in 2001 as one of the first immune systems approaches to classification. It is a supervised learning paradigm based on the principles of resource-limited artificial immune systems [6], [7]. In 2002 Watkins and Timmis [8] suggest improvements to AIRS algorithm that are capable of maintaining classification accuracy, whilst improving performance in terms of computational costs and an increase in the data reduction capabilities of the algorithm. This algorithm is here named AIRS2. A new version for a parallel AIRS2 was present in 2004 to explore ways of exploiting parallelism inherent in an artificial immune system for decreased overall runtime [9]. This algorithm was used in the present paper.

The classification results using those two immune techniques were compared with classification elaborated by a Petroleum Engineering expert [10] and with others automated methods in solving of drilling operation stages identification problem. These methods are neural networks [11], Support Vector Machines (SVM) [12] and Locally Weighted Learning (LWL) [13].

2 Mud-Logging System

During the petroleum well drilling operation many mechanical and hydraulic parameters are measured and monitored in order to perform the drilling in a safe and optimized manner. There are many systems that work together in a rig to accomplish this task. One of these systems is called a mud-logging system and it is responsible for measuring and monitoring a set of mechanical and geological parameters.

Mud-logging system techniques were introduced in Brazil in the 80's. At that time, only a reduced number of parameters were monitored. Since the 80s, with the developments in instrumentation techniques, the number of measured parameters has increased and the use of mud-logging systems became a common practice in the oil industry.

Another aspect that contributed to the progress of mud-logging techniques in Brazil was the development of deep and ultra-deep water drilling technologies. The deep and ultra-deep-water environments require a more controlled drilling operation [14]. Any failure or inattention may cause great human and economic losses. In order to have a more controlled process, the information supplier systems needed to be improved. In this context, the mud-logging systems were enhanced to become an important information supplier system.

Nowadays, mud-logging systems have two distinctive dimensions, the first one is responsible for collecting and analyzing formation samples (shale-shaker samples), and the second one is responsible for measuring and monitoring mechanical parameters related to the drilling operation. Considering only the second dimension, the mud-logging system could be characterized as a complete instrumentation system.

To accomplish its mission, the mud-logging systems rely on a wide range of sensors distributed in the rig operative systems. One important characteristic of this technique is that there is no sensor inside the well, and all measurements are taken on the rig. The data collect by the sensors are sent to a central computer system, where the data are processed and displayed in real time through screens installed in the mud-logging cabin and in the company-man office. The checking of the parameter evolution is carried out using the monitors; the system not only permits the selection of the displayed parameters but also the selection of their presentation appearance (numbers or graphics). Throughout the whole drilling operation, there is a worker watching the parameters for any kind of abnormality. If an observed parameter presents an unusual behavior, the worker has to immediately communicate this to the driller that will carry out the appropriate procedures to solve the problem. In fact, the system permits the programming of alarms that will sound in the mud-logging cabin, alerting the mud-logging worker, always when the value of the observed parameter is not within the programmed range.

The number of observed parameters may vary according to the particular characteristic of the drilling operation. The most common measured parameters are: Well Depth (Depth), True Vertical Depth (TVD), Bit Depth, Rate of Penetration (ROP), Hook Height, Weight on Hook (WOH), Weight on Bit (WOB), Vertical Rig Displacement (Heave), Torque, Drillstring rotation per minute (RPM), Mud Pit Volume, Pump Pressure, Choke Line Pressure, Pump Strokes per minute (SPM), Mud Flow, Total Gas, Gas Concentration Distribution, H2S concentration, Mud Weight in/out, Drilling Fluid Resistivity, Drilling Fluid Temperature, Flow Line, LAG Time, and Stand Length.

It is important to mention that just some of the listed parameters are really measured using sensors. Some of them are calculated using the measured parameters. The WOB, for instance, is a calculated parameter. It is calculated using the WOH (a measured parameter) and the knowledge of the weight of drill string elements.

The mud-logging monitoring services are generally provided by a specialized company that, at the end of the drilling operation, makes a report relating the occurrences associated to the completed operation. During the drilling monitoring, a huge amount of data is generated, and due to difficulties of data storage, the data are summarized to make up smaller files. The common practice is to reduce measurements made on a second basis to measurements made on a minute basis. Although it solves the problem of the files volume, on the other hand it represents the loss of a large amount of information. There are some events that may occur and last only a few seconds, like the drag occurrence in tripping out. When the data is summarized, the information about the drag occurrence is partially lost.

Considering all the measured parameters, it can be noted that the parameters related to the gas invasion in the well (Mud Pit Volume and Total Gas) are used more often than the others, it indicates that there is still a great information potential that has not been properly used.

Another important question related to the mud-logging system is the redundancy in parameter measuring. Besides hook height, other parameters have been measured by more than one instrument system. It is common to find rigs where the same parameter is being measured by the mud-logging company, by the MWD company and by the rig itself. And it is not rare to observe cases where the three measurements taken do not present the same absolute value. This behavior has caused some questioning about the future of mud-logging systems.

The general tendency is that more modern rigs will have a higher level of instruments on their working systems, and maybe in the future the rig will be in charge of measuring and monitoring all drilling parameters while the mud-logging services will be restricted to shale-shaker sample analyses.

3 Individual Stages Associated to the Drilling Operation for the Classification System

The drilling of petroleum well is not a continuous process made up of one single operation. If one looks at in a minor scale, it is possible to note that the petroleum well drilling operation is made up of a sequence of discrete events. These minor events comprised into the drilling operation will be called drilling operation stages. Six basics stages associated to the drilling operation were identified to build the proposed classification system. A brief description of each considered stage is presented below:

- **Rotary Drilling:** in this stage the drilling itself occurs, the bit really advances increasing well depth. The drill string is rotating and there is mud circulation. The drill string is not anchored to the rotary table causing a high hook weight level.
- **Rotary Reaming:** in this stage despite the high hook weight level, mud circulation and drillstring rotation, the bit does not advance increasing the final well depth. In this situation, there is a back-reaming of an already drilled well section.
- **Oriented Drilling ("Sliding Drilling"):** in this stage, the bit really advances increasing the well depth. The difference here is that the drillstring is not rotating and the drilling occurs due to the action of the downhole motor. There is mud circulation and a high hook weight level.
- **Back-reaming or Tool adjusting:** in this stage, the bit does not advance increasing the final well depth. There is circulation and a high hook weight level. This condition indicates that back-reaming is being carried out or that the tool-face of the downhole tool is being adjusted.
- **Tripping:** this stage corresponds to the addiction of a new section to the drill-string. The drillstring is anchored causing a low hook weight level. The drill string dos not rotate.
- **Circulating:** in this stage there is no gain in the well depth. It is characterized by fluid circulation, a high hook weight level and a moderated rotation of the drillstring.

These six stages represent a first effort to individualize the basic components of a drilling operation. The stages were detailed considering the drilling phases with mud return to the surface. The drilling technology considered was the drilling using mud motor and bent housing. This classification may not be satisfactory for the initial drilling phases and for special operations, such as fishing, in the well. In the same way, if other drilling technologies are considered, like the rotary steerable systems, small adjustments in the definition of the stages will be required. For instance, when using rotary steerable systems, it makes no sense to make a distinction between rotary drilling and oriented drilling stages as they were defined in this work, because these systems are supposed to drill all the time using drillstring rotation.

4 Artificial Immune Systems

The immune system guards our bodies against infections due to the attacks of antigens. The natural immune system offers two lines of defense, the innate and adaptive immune system. The innate immune system consists of cells that can neutralize a predefined set of attackers, or 'antigens', without requiring previous exposure to them. The antigen can be an intruder or part of cells or molecules of the organism itself. This part of the immune system is not normally modeled by AIS systems.

Vertebrates possess an adaptive immune system that can learn to recognize, eliminate and remember specific new antigens. This is accomplished by a form of natural selection. The adaptive immune response in biological systems is based on two kinds of lymphocytes (or self-cells) in the body: T-cells, so named because they originate in the thymus gland, and B-cells, which originate in bone marrow [3].

The major responsibility of the T-cells and B-cells is the secretion of the receptors called the antibodies (*Ab*) as a response to the antigens that enter the body (*Ag*) (nonself-cells). The role of these receptors on the surface of the lymphocytes is to recognize and bind the antigen. An individual T-cell or B-cell responds like a pattern matcher - the closer the antigen on a presenting cell is to the pattern that a T-cell or B-cell recognizes, the stronger the *affinity* of that T-cell or B-cell for the antigen. T-cells are sometimes called helper T-cells because in nature, although the B-cells are the immune response mechanism that multiplies and mutates to adapt to an invader, it is only when a T-cell and B-cell respond together to an antigen that the B-cell is able to begin cloning itself and mutating to adjust to the current antigen ('clonal expansion' or 'clonal selection') [15].

Once a B cell is sufficiently stimulated though close affinity to a presented antigen, it rapidly produces clones of itself. At the same time, it produces mutations at particular sites in its gene which enable the new cells to match the antigen more closely. There is a very rapid *proliferation* of immune cells, successive generations of which are better and better matches for the antigens of the invading pathogen. B cells which are not stimulated because they do not match any antigens in the body eventually die [16].

The immediate reaction of the innate and adaptive immune system cells is called the primary immune response. A selection of the activated lymphocytes is turned into sleeper memory cells that can be activated again if a new intrusion occurs of the same antigen, resulting in a quicker response. This is called the secondary immune response. Interestingly, the secondary response is not only triggered by the re-introduction of the same antigens, but also by infection with new antigens that are similar to previously seen antigens. That is why we say that the immune memory is *associative*.

Artificial Immune System (AIS) are inspired in many aspects of the natural immune systems, such as adaptivity, associative memory, self/non-self discrimination, competition, clonal selection, affinity maturation, memory cell retention, mutation and so on. These artificial immune system algorithms (also known as immunocomputing algorithms) have been applied to a wide range of problems such as biological modeling, computer network security, intrusion detection, robot navigation, job shop scheduling, clustering and classification (pattern recognition). We are interested in this last kind of application for our problem of classifying the well drilling stages. We have considered the two most known classification algorithms based on immune systems to carry out this task: CLONALG and AIRS2.

4.1 Clonal Selection Algorithm (CLONALG)

The clonal selection algorithm, CSA, was first proposed by de Castro and Von Zuben in [4] and was later enhanced in their 2001 paper [5] and named CLONALG. It uses the clonal selection principle to explain the basic features of an adaptive immune response to an antigenic stimulus. It establishes the idea that only those cells that recognize the antigens are selected to proliferate. The selected cells are subject to an affinity maturation process, which improves their affinity to the selective antigens. The algorithm takes a population of antibodies and by repeated exposure to antigens, over a number of generations, develops a population more sensitive to the antigenic stimulus. The basic algorithm for pattern recognition is [5]:

1. Randomly generate an initial population of antibodies *Ab*. This is composed of two subsets *Abm* (memory population) and *Abr* (remaining population): $Ab = Abm \cup Abr \ (m + r = N)$.
2. Create a set of antigenic patterns *Ag*.
3. Randomly choose an antigen Ag_i from the population *Ag*.
4. For all the *N* antibodies in *Ab* calculate its affinity f_i to the antigen Ag_i using some affinity function (Hamming Distance).
5. The *n* selected antibodies will be cloned (reproduced) independently and proportionally to their antigenic affinities, generating a repertoire C_i **o**f clones: the higher the antigenic affinity, the higher the number of clones generated for each of the *n* selected antibodies.
6. The repertoire C_i is submitted to an affinity maturation process inversely proportional to the antigenic affinity, generating a population C_i^* of matured clones: the higher the affinity, the smaller the mutation rate.
7. Re-apply the affinity function f_i to each member of the population C_i^* and select the highest score as candidate memory cell *Abm*. If the affinity of this antibody with relation to Ag_i is greater than the current memory cell Abm_i, then the candidate becomes the new memory cell.
8. Remove those antibodies with low affinity in the population *Abr*. Finally, replace the *d* lowest affinity antibodies from *Abr*, with relation to Ag_i, by new randomly generated individuals.
9. Repeat steps 3-8 until all *M* antigens from *Ag* have been presented.

A *generation* is completed after performing the steps 3-9 above. The rate of clone production is decided using a ranking system. Mutation can be implemented in many ways, such as multi-point mutation, substring regeneration and simple substitution.

4.2 Parallel Artificial Immune Recognition System – Version 2 (Parallel AIRS2)

AIRS2 is a bone-marrow, clonal selection type of immune-inspired algorithm. AIRS2 resembles CLONALG in the sense that both algorithms are concerned with developing a set of memory cells that give a representation of the learned environment. AIRS2 also employs affinity maturation and somatic hypermutation schemes that are similar to what is found in CLONALG. AIRS2 has used population control mechanisms and has adopted use of an affinity threshold for some learning mechanisms.

AIRS2 is concerned with the discovery/development of a set of memory cells that can encapsulate the training data. Basically, this is done in a two-stage process of first evolving a candidate memory cell and then determining if this candidate cell should be added to the overall pool of memory cells [8]. This process can be outlined from [9] as follows:

1. *Initialization*: Create a random base called the memory cells pool.
2. *Clonal expansion*. Compare a training instance with all memory cells of the same class and find the memory cell with the best affinity (Euclidian distance) for the training instance. We will refer to this memory cell as mc_{match}.
3. *Affinity maturation*. Clone and mutate mc_{match} in proportion to its affinity to create a pool of abstract B-Cells.
4. *Metadynamics of B-Cells*. Calculate the affinity of each B-Cell with the training instance.
5. Allocate resources to each B-Cell based on its affinity.
6. Remove the weakest B-Cells (lowest affinity) until the number of resources returns to a preset limit.
7. *Cycle*. If the average affinity of the surviving B-Cells is above a certain level, continue to step 8. Else, clone and mutate these surviving B-Cells based on their affinity and return to step 4.
8. *Metadynamics of memory cells*. Choose the best B-Cell as a candidate memory cell (mc_{cand}).
9. If the affinity of mc_{cand} for the training instance is better than the affinity of mc_{match}, then add mc_{cand} to the memory cell pool. If, in addition to this, the affinity between mc_{cand} and mc_{match} is within a certain threshold, then remove mc_{match} from the memory cell pool.
10. *Cycle*. Repeat from step 2 until all training instances have been presented.

Once this training routine is complete, AIRS2 classifies instances using *k*-nearest neighbor (k-NN) with the developed set of memory cells.

Comparing with a data mining approach, AIRS2 is a cluster-based procedure to classification. It first learns the structure of the input space by mapping a codebook of cluster centers to it and then uses k-nearest neighbor on the cluster centers for classification. The attractive point of AIRS2 is its supervised procedure for discovering both the optimal number and position of the cluster centers.

Algorithmically, based on the above description, the parallel version of AIRS2 behaves in the following manner [9]:

a) Read in the training data at the root process.
b) Scatter the training data to the *np* processes.
c) Execute, on each process, steps 1 through 9 from the serial version of AIRS2 on the portion of the training data obtained.
d) Gather the developed memory cells from each process back to the root.
e) Merge the gathered memory cells into a single memory cell pool for classification.

5 Results

The classification problem consists in identifying the drilling operations described above as *Rotary Drilling* (RD), *Rotary Reaming* (RR), *Oriented Drilling ("Sliding Drilling")* (SD), *Back-reaming or Tool adjusting* (TA), *Tripping* (TR) and *Circulating* (CI).

In order to identify a given drilling stage in execution, the system needs some of the information monitored by the mud-logging system. This work uses: Bit Depth, Weight on Hook (WOH), Stand Pipe Pressure (SPP), Drillstring Rotation (RPM) and Weight on Bit (WOB) for this task, as in the previous works [11], [12].

Real records of mud-logging data consisting of 3784 samples of three days well drilling were used for the training and evaluation of the implemented immune classifier. A Petroleum Engineering expert classified previously these data [10]. When training AIS classifier, the whole data set (3784 samples) was randomly separated into two subsets: 75% as training set (2838 samples) and 25% as testing set (946 samples) after training. These sets were the same used in the others related classification methods for this problem [11], [12]. Table 1 shows the data distribution according to pre-defined classes for the training and test sets. The table clearly indicates the data imbalance issue among the classes, mainly for the Circulating (CI) and Tripping (TR) stages, which are the less usual operation in the drilling activity.

Table 1. Distribution of data per class in the training and test sets

Number of samples	**Drilling operations**						
	CI	**TR**	**TA**	**SD**	**RR**	**RD**	**Total**
Training Set	14	75	795	753	343	858	*2838*
Test Set	2	22	266	253	114	289	*946*

The application of CLONALG with 20 generations for the proposed task produced 400 incorrectly identified instances of the training set and 104 misclassified instances of the test set. The classification accuracy for training and test sets are 85.9% and 89.0%, respectively. Table 2 shows the correctness rate for the training and test samples for each evaluated method.

Parallel AIRS2's experiments are undertaken with the *k*-value for the *k* nearest neighbor approach is set to 7. The value for number of threads is 5. The learning evaluation of this approach has shown a reasonable performance, obtaining 2587 instances correctly classified (91.2%) for the training data and 879 instances (92.9%) for the test data. The performance of the Parallel AIRS2 is higher than of CLONALG.

A Multi-Layer Perceptron (MLP) neural network (NN) with backpropagation (BP) learning algorithm, which is widely used in numerous classification applications, has been investigated for this problem in our previous work [11] and its results are compared with the immune classifier systems CLONALG and Parallel AIRS2. MPL-BP has a better performance among all method, reaching an accuracy of 96.3% and 94.9% for the training and test sets, respectively.

Support Vector Machine (SVM) was used also to develop the automatic classification system of well drilling stages [12]. SVM correctly classified 2660 samples of the training set, reaching a reliability of 93.73%. For the testing set, 876 samples were well classified, with 92.6% of success.

Locally weighted learning (LWL) is a class of statistical learning techniques (*lazy learning*) that provides useful representations and training algorithms for learning about complex phenomena [13]. LWL uses locally-weighted training to combine training data, using a distance function to fit a surface to nearby points. It is used in conjunction with another classifier to perform classification rather than prediction.

Table 2. Correctness rate for each classification method

Method	Training Set	Test Set
MLP-BP	96.3%	94.9%
SVM	93.7%	92.6%
CLONALG	85.9%	89.0%
Parallel AIRS2	91.2%	92.9%
Lazy LWL	80.5%	81.3%

Table 3. Classification accuracy for each class in the training data

Method	Drilling operations					
	CI	TR	TA	SD	RR	RD
MLP-BP	100%	100%	97.5%	98.1%	89.2%	96.0%
SVM	100%	100%	96.6%	95.8%	83.1%	92.8%
CLONALG	0%	98.7%	85.0%	93.5%	45.5%	96.5%
Parallel AIRS2	57.1%	100%	90.1%	95.6%	69.7%	96.6%
Lazy LWL	0%	0%	91.7%	0%	91.5%	100%

Table 4. Classification accuracy for each class in the test data

Method	Drilling operations					
	CI	TR	TA	SD	RR	RD
MLP-BP	100%	100%	96.2%	98.4%	86.0%	93.8%
SVM	100%	100%	96.6%	94.1%	83.3%	90.7%
CLONALG	0%	100%	89.5%	96.0%	72.8%	97.2%
Parallel AIRS2	100%	100%	91.7%	97.6%	75.4%	98.3%
Lazy LWL	0%	0%	91.7%	0%	93.3%	100%

The four components that define LWL are: a distance metric, near neighbors, weighting function, and fitting the local model. In this application it is the technique with the worst result: precision of 80.5% for training set and 81.3% for test set.

In order to understand the difficulties of pattern discrimination of each method in the learning process, Tables 3 and 4 present the classification accuracy for each class related to each learning technique for both training and test data. Ten trial runs were performed for each method using a 10-fold cross-validation procedure.

It is important to mention that the circulating (CI), tripping (TR), rotating mode (consisted of rotary drilling (RD) and rotary reaming (RR) stages) and non-rotating mode (consisted of back-reaming or tool adjustment (TA) and sliding drilling (SD) stages) operations are linearly separable classes. However, RD and RR are non-linearly separable classes. The same for TA and SD classes.

Close examination of the Tables 3 and 4 revels that, as expected, for MLP-BP and SVM the accuracy on the non-linearly separable data set is less than the classification accuracy on the linearly separable dada set. The classification is well successful either for classes with a little amount of samples or a large one.

When CLONALG and Parallel AIRS2 algorithms use the imbalanced data set for training, antigens from the majority class may generate more memory cells than the ones from the minority class. If all the memory cells are represented in a high dimensional space, one minority class cell may be surrounded by many majority class cells; so taking votes (k-NN classification) from several memory cells closest to a test antigen may cause biased decisions. That explains the fact of CLONALG is unable to learn the CI class and Parallel AIRS2 has a low performance to this class. Nevertheless, even for complex mud-logging data sets CLONALG and Parallel AIRS2 algorithms are able to perform fairly well as a classifier.

LWL does not learn the minority classes and its behavior for SD class is unclear.

6 Conclusion

The classification systems presented can be used either to classify stored mud-logging data of a database of drilled wells or to classify mud-logging data on-line and online in a rig. Due to the detailed level regarding each executed stage provided by the classification systems, it can help to analyze the individual drilling performance of each well. Information about the total time spent on each stage combined with related economic costs can be used to assess the real cost reduction benefit caused by optimized drilling programs and introduction of new technologies.

The imbalanced real mud-logging data has a large impact on the classification performance of the AIS classifiers, since they can achieve high precision on predominant classes but very low correct classification on classes with a few samples, in contrast with the neural network and SVM, which recognize efficiently all patterns of the minority classes. The results suggest that Parallel AIRS2 could achieve a similar performance than MLP-BP and SVM do on data sets of others applications or problems with a better class's distribution.

This paper demonstrates that the development of a classification system for real multi-class problems using immune system inspired approaches is feasible.

Acknowledgements

The authors would like to thank the Brazilian Research Council (CNPq) for the financial support (Process CT-Petro/MCT/CNPq 504528/2004-1).

References

1. Unneland, T., Hauser, M.: Real-Time Asset Management: From Vision to Engagement–An Operator's Experience. In: Proc. SPE Annual Technical Conference and Exhibition, Dallas, USA (2005)
2. Yue, Z.Q., Lee, C.F., Law, K., Tham, L.G.: Automatic monitoring of rotary-percussive drilling for ground characterization – illustrated by a case example in Hong Kong. International Journal of Rock Mechanics & Mining Sciences 41, 573–612 (2003)

3. de Castro, L., Timmis, J.: Artificial immune systems: A new computational approach. Springer, London. UK (2002)
4. de Castro, L.N., Von Zuben, F.J.: The Clonal Selection Algorithm with Engineering Applications. In: Whitley, L.D., Goldberg, D.E., Cantú-Paz, E., Spector, L., Parmee, I.C., Beyer, H.-G. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '00), Workshop on Artificial Immune Systems and Their Applications, pp. 36–37. Morgan Kaufmann, Las Vegas, Nevada, USA (2000)
5. 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)
6. Watkins, A., Boggess, L.: A new classifier based on resource limited Artificial Immune Systems. In: Proc. 2002 Congress on Evolutionary Computation (CEC2002), Honolulu, Hawaii. IEEE Press, New York (2002)
7. 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)
8. Watkins, A., Timmis, J.: Artificial Immune Recognition System (AIRS): Revisions and Refinements. In: Timmis, J., Bentley, P.J. (eds.) Proc. of 1st International Conference on Artificial Immune Systems (ICARIS2002), pp. 173–181. University of Kent at Canterbury (2002)
9. Watkins, A., Timmis, J.: Exploiting 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)
10. Tavares, R.M., Mendes, J.R.P., Morooka, C.K., Plácido, J.C.R.: Automated Classification System for Petroleum Well Drilling using Mud-Logging Data. In: Proc. of 18th International Congress of Mechanical Engineer.Offshore & Petroleum and Engineering, Ouro Preto, Brazil (2005)
11. Serapião, A.B.S., Tavares, R.M., Mendes, J.R.P., Morooka, C.K: Classificação automática da operação de perfuração de poços de petróleo através de redes neurais. In: Proc. of VII Brazilian Symposium on Intelligent Automation (SBAI). São Luís-MA, Brazil (2005)
12. Serapião, A.B.S., Tavares, R.M., Mendes, J.R.P., Guilherme, I.R.: Classification of Petroleum Wells Drilling Operations Using Support Vectors Machine (SVM). In: Proc. of International Conference on Computational Intelligence for Modelling, Control and Automation (CIMCA'2006). IEEE Computer Society, Sydney, Australia (2006)
13. Atkeson, C.G., Moore, A.W., Schaal, S.: Locally weighted learning. Artificial Intelligence Review 11(1-5), 11–73 (1997)
14. Kyllingstad, A., Horpestad, J.L., Klakegg, S., Kristiansen, A., Aadnoy, B.S.: Factors Limiting the Quantitative Use of Mud-Logging Data. In: Proc. of the SPE Asia Pacific Oil and Gas Conference, Singapore (1993)
15. Ada, G.L., Nossal, G.J.V.: The Clonal Selection Theory. Scientific American 257(2), 50–57 (1987)
16. Berek, C., Ziegner, M.: The Maturation of the Immune Response. Imm. Today 14(8), 400–402 (1993)

The SUPRAIC Algorithm: A Suppression Immune Based Mechanism to Find a Representative Training Set in Data Classification Tasks

Grazziela P. Figueredo[1], Nelson F.F. Ebecken[1], and Helio J.C. Barbosa[2]

[1] COPPE/UFRJ, Rio de Janeiro, RJ, Brazil
[2] LNCC/MCT, Av. Getúlio Vargas 333
25651 070 Petrópolis RJ, Brazil

Abstract. This article proposes a new classifier inspired on a biological immune systems' characteristic. This immune based predictor also belongs to the class of k-nearest-neighbors algorithms. Nevertheless, its main features, compared to other artificial immune classifiers, are the assumption that training set is the antibodies' population and a suppression mechanism that tries to reduce the training set into a smaller subset. This subset is supposed to contain the most significative samples, without loosing much capability of generalization. It is known that in prediction problems, the choice of a good training set is crucial for the classification process. And this is the focus of this research. Experiments using some benchmarks and the analysis of the results of our ongoing work are presented.

1 Introduction

Classification [1] and pattern recognition are important tasks in all knowledge fields [2]. To classify means to categorize into a finite number of classes elements defined by a group of attributes. There are many types of statistical and artificially intelligent classifiers, as it can be seen in [3,4]. One of the main issues in classification problems involves the choice of good samples to train a classifier. A training set capable to represent well the characteristics of a class has better chances to establish a successful predictor.

This article proposes a new classifier inspired on biological immune systems' characteristics. The model belongs to the class of k-nearest-neighbors algorithms. Its main features, compared to other artificial immune classifiers [5,6,7,8,9,10,11,12,13,14,15,16,13,17,18,19,20,21,22], are the assumption that the training set constitutes the initial antibodies' population of the system, and a suppression mechanism that tries to reduce this training set into a smaller subset. This subset is supposed to contain the most significative samples, without loosing much capability of generalization.

In this proposal, the inspiration came from the self-regulation mechanism from the biological immune systems. Clones of B cells that are no longer needed

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 59–70, 2007.

in the organism suffer apoptosis. Therefore, another feature of the algorithm that distinguish it from the others mentioned, is its simplicity. There are no mechanisms such as affinity maturation, clonal selection [23,24], its not based on imune network models or kohonen networks [19].

1.1 The Behavior of Biological Immune Systems

According to the IS self regulation mechanism, clones no longer needed by the organism, or those that self attack, do not receive signals to keep alive and suffer apoptosis. Those signals came from the lymph nodes or T helper cells [25]. This self-regulation characteristic allows the organism to save energy and keep only the repertory of lymphocytes really needed for self defense. These concepts are the inspiration of the presented model.

Basically, the algorithm presented here draws from a concept in which the model for the system should evolve to produce antibodies to recognize the training data and be capable to identify new presented antigens. Instead of working with a system that generates and evolves clones of B cells until the antibodies recognize the training group, this paper proposes the training data to be the very same antibodies' repertory of the system.

The suppression concept is employed in the training set to eliminate very similar antibodies. For suppression, the antigens, or test data, are divided into two subgroups. The first will be responsible for testing the model and eliminating redundant antibodies. The second subgroup is used to validate the efficiency of the remaining antibodies after suppression.

This article is organized as follows. Section 2 presents a detailed description of the algorithm proposed together with the suppression mechanism. Experiments and results obtained from standard databases found in the literature are presented. Finally, the last section presents the final remarks on the model and propose some suggestions for further research.

2 The Proposed Algorithm

The algorithm starts with the idea that the system's model must evolve to create antibodies that recognize the training set and be able to identify new presented antigens. Therefore, instead of the system generating and evolving B cells clones until the antibodies recognize the training set and establish a cellular memory, it is proposed that the training set itself constitutes the repertory of antibodies of the system. It means that the immunological memory represented by Artificial Recognition Balls (ARBs), which are mathematical metaphors for antibodies or B-cells [26,27,24], (see Figure 1), will be ready for generalization from the moment training data is placed in the system. In the presented figure, S is the shape space where • are the antibodies, × are the antigens and ε denotes the radius of each antibody [27,24].

The set of attributes is normalized to belong to the same scale of values. In this case, the data was transformed to fit in the interval [0,1]. At the present moment of this research, only real or integer data are considered.

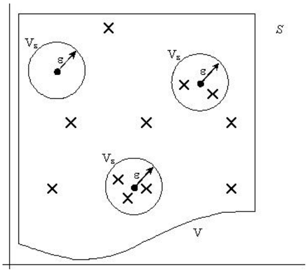


Fig. 1. The shape space model

To proceed the system's learning, the database is divided into three subsets, training representing the antibodies, testing and validating as antigens. The initial proportion of samples adopted for each group, respectively, was 60%, 20% and 20% and then, another experiment with the proportions 70%, 20% and 10%.These proportions were adopted empirically. Both antigens and antibodies are represented by an array containing the attributes. The antigens, or test data, are classified according to the closest antibody. It means that in this case, instead of a shape space with variant ε, the model uses the Voronoi diagram [28], that can be seen in Figure 2. In the figure, dots are the centroids of the ARBs and represent the antibodies that cover a certain group of nearest antigens.

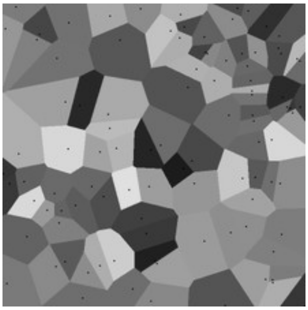


Fig. 2. Voronoi diagram: a representation of the classification process

The closest antibody is determined by a measure of distance. In the experiments, the adopted ones were the Euclidian and Manhattan distances, described respectivelly by the equations 1 and 2.

$$distance = \sqrt{\sum_{i=1}^{TotalAttributes} (Antibody[i] - Antigen[i])^2} \qquad (1)$$

$$distance = \sum_{i=1}^{TotalAttributes} |Antibody[i] - Antigen[i]| \qquad (2)$$

The reason why antigens have been split into two subsets is to provide the suppression mechanism. The suppression concept is used among the training set, so that very similar antibodies are eliminated and the best ones are kept, following the theories of self-regulation and affinity maturation of biological immune systems, described in section 1.1. In other words, those antibodies able to recognize antigens from the test set remain while the others are eliminated from the population.

In this artificial classifier, these signals for survival are represented by a counter variable for each antigen recognized by an antibody, independently if the classification is correct or not. It is known that some databases have limits on the percentage of correctly classified samples. This is the reason why antibodies that classify incorrectly are still maintained on the suppressed population. These antibodies could be viewed as cross-reactive ones.

The schematic algorithm can be seen in Algorithm 1, called SUPRAIC.[1]

Algorithm 1. Algorithm SUPRAIC

```
◇ Read the database file;
◇ Normalize data between the interval [0,1];
◇ Determine antibodies and antigens by dividing the data (in a uniform distribution) into, for example, 60% antibodies and 40% antigens;
◇ For each antibody, set its counter variable = 0;
◇ Divide antigens into test and validation subgroups. Generally, it is used the same proportion for each one, 20%;
◇ Determine the inicial antibodies as the training set.
◇ Train the model via test subset by finding the nearest antibody for each antigen. The nearest antibody is measured by the Euclidian distance among attributes;
◇ For each antigen recognized by the antibody, increase its counter variable by one;
◇ Suppress the remaining antibodies not capable to recognize (counter = 0) antigens. Eliminate them from the antibodies population.
◇ Validate the final predictor, already suppressed by the previous step, by using the remaining antibodies to recognize the validation set;
◇ Calculate the accuracy of the final predictor.
```

3 Experiments and Results

This section presents experiments using some benchmark examples of databases extracted from the UCI machine learning repository [29]. The metrics adopted to evaluate the efficiency of the classifier were extracted from [2].

For two classes problems, the metrics are based on confusion matrix, a tool which informs the sorts of hits and errors made by a classifier [2]. The classes are named positive and negative and the confusion matrix has four values computed in terms of real and predicted classes, namely:

- TP (true positives): the amount of positive elements predicted as positive;
- TN (true negatives): the amount of negative elements predicted as negative;

[1] Suppressor Artificial Immune Classifier.

- FP (false positives): the amount of negative elements predicted as positive;
- FN (false negatives): the amount of positive elements predicted as negative;

With the values above determined, the most common metrics that will be used to determine the efficiency of the proposed predictor are:

- Accuracy (acc) and Validation (val): they are the ratio of correct decisions made by a classifier. The difference between both is that the first one is used to train the predictor and the second one is to validate it:

$$acc(val) = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

- Sensitivity (sens): it measures how much a classifier can recognize positive examples:

$$sens = \frac{TP}{TP + FN} \tag{4}$$

- Specificity (spec): it measures how much a classifier can recognize negative examples:

$$spec = \frac{TN}{TN + FP} \tag{5}$$

- Precision (prec): it is the ratio of predicted positive examples which really are positives:

$$prec = \frac{TP}{TP + FP} \tag{6}$$

- F-measure (FMea): it is the harmonic mean of sensitivity and precision. In this study, the parameter β was set to zero:

$$prec = \frac{(\beta^2 + 1) \times sens \times prec}{sens + \beta \times prec} \tag{7}$$

- G-mean (GSP): it is the geometric mean of sensitivity and precision:

$$GSP = \sqrt{sens \times prec} \tag{8}$$

- G-mean2 (GSS): it is the geometric mean of sensitivity and specificity:

$$GSS = \sqrt{sens \times spec} \tag{9}$$

In all the experiments that follow, β was set to zero, which means that sensitivity and precision have the same importance.

Table 1. Results from the predictor without suppression mechanism for Breast Cancer Database with 419 antibodies

percentage split	distance	acc	val	sens	spec	prec	Fmea	GSP	GSS
60-20-20	Euclidian	0.96	0.97	0.94	0.99	0.98	0.48	0.96	0,96
60-20-20	Manhattan	0.96	0.98	0.94	1	1	0.47	0.95	0.95
70-20-10	Euclidian	0.97	0.97	0.93	1	1	0.48	0.96	0.96
70-20-10	Manhattan	0.98	0.94	0.85	1	1	0.46	0.92	0.92

Table 2. Results from the predictor with suppression mechanism for Breast Cancer Database with 56 antibodies

percentage split	distance	acc	val	sens	spec	prec	Fmea	GSP	GSS
60-20-20	Euclidian	0.96	0.94	0,85	0.99	0.98	0.45	0,91	0,91
60-20-20	Manhattan	0.98	0.96	0.88	1	1	0.47	0.94	0.94
70-20-10	Euclidian	0.97	0.96	0.89	1	1	0.47	0.94	0.94
70-20-10	Manhattan	0.98	0.96	0.88	1	1	0.47	0.94	0.94

Table 3. Results from other classifiers to cancer database

classifier	acc
J48 (Decision Tree)	0.95
Multi-Layer Perceptron	0.97
Naive Bayes	0.98

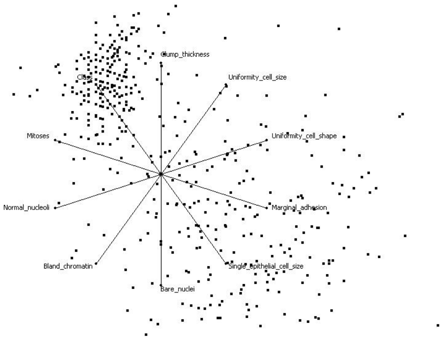


Fig. 3. The total amount of antibodies (training set) for the Cancer database

3.1 Breast Cancer Database

The purpose of this work was to reduce training sets without loosing significantly the accuracy of the predictor. The first experiment of this new proposal was made on the breast cancer database. The breast cancer database is characterized for having 699 registers, 9 attributes and 2 classes.

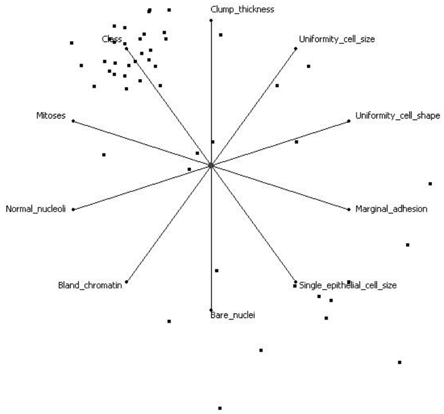


Fig. 4. Antibodies from Cancer database after suppression and 60-40% split

Table 1 shows the results with the previously mentioned metrics, but using the classifier without the suppression mechanism. In Table 2, the suppression mechanism was applied.

As it can be seen, comparing the two results in Tables 1 and 2, there was not too much difference between the metrics. Nevertheless, the amount of antibodies was reduced from 419 to 56. That means a reduction of almost 87% of training data. Figure 3 shows training data before suppression and Figure 4 shows the final suppressed classifier using star coordinates.

3.2 Pima Indians Diabetes Database

Database diabetes is characterized by 768 registers, 8 attributes and 2 classes. As in the first database studied, it also can be seen that the results from Tables 4 and 5 are not very different, except for the value specificity. Even though, the reduction of the antibodies by the suppression was from 460 to 80. Figures 5 and 6 shows, respectively, the training data before suppression and data already suppressed with the most significant antibodies. Table 6 shows results obtained from other common classifiers: Naive Bayes, Multi-Layer Perceptron and Decision Tree [3].

Table 4. Results from the predictor without suppression mechanism for Pima Indians Diabetes Database with 460 antibodies

percentage split	distance	acc	val	sens	spec	prec	Fmea	GSP	GSS
60-20-20	Euclidian	0.74	0.67	0.7	0.62	0.77	0.36	0.73	0.66
60-20-20	Manhattan	0.71	0.69	0.71	0.65	0.79	0.37	0.75	0.68
70-20-10	Euclidian	0.70	0.69	0.74	0.61	0.74	0.37	0.74	0.67
70-20-10	Manhattan	0.73	0.71	0.74	0.68	0.77	0.38	0.76	0.71

Table 5. Results from the predictor with suppression mechanism for Pima Indians Diabetes Database with 80 antibodies

percentage split	distance	acc	val	sens	spec	prec	Fmea	GSP	GSS
60-20-20	Euclidian	0.74	0.65	0.77	0.45	0.71	0.37	0.74	0.59
60-20-20	Manhattan	0.71	0.65	0.8	0.4	0.7	0.37	0.74	0.56
70-20-10	Euclidian	0.70	0.65	0.72	0.55	0.70	0.35	0.71	0.63
70-20-10	Manhattan	0.73	0.66	0.80	0.45	0.68	0.37	0.74	0.60

Table 6. Results from other classifiers to diabetes database

classifier	acc
J48 (Decision Tree)	0.76
Multi-Layer Perceptron	0.75
Naive Bayes	0.70

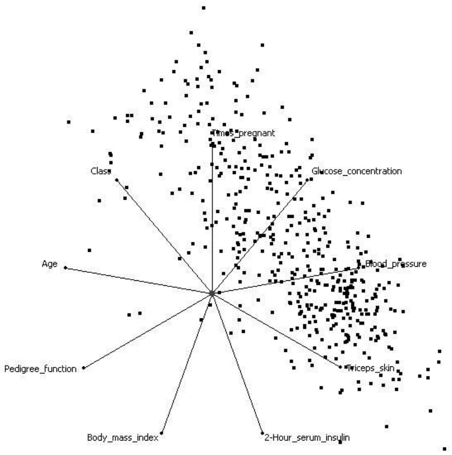


Fig. 5. The total amount of antibodies (training set) for the Diabetes database

3.3 Iris Database

The following database is characterized for having more than two classes. Iris database has 150 registers, 4 attributes and 3 classes. Although there are some other metrics to evaluate multi-class problems [2], in this work it will be adopted only accuracy on test and validation sets. As a continuation of this work, it is intended to use metrics proposed by [2].

In Tables 7 and 8 are shown the results for the iris database. Although in the first case the accuracy and validation test are classified, respectively, as 97% and 100% correct and in Table 8 shows an inferior performance, the decrease of the antibodies is significant – from 90 to 10.

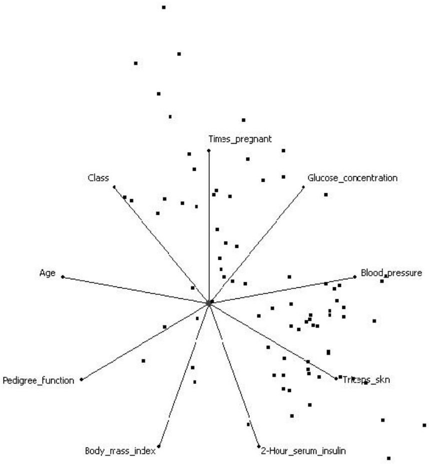


Fig. 6. Antibodies from Diabetes database after suppression and 60-40% split

Table 7. Results for the Iris Database with 90 antibodies without suppression

percentage split	distance	acc	val
60-20-20	Euclidian	0.97	1
60-20-20	Manhattan	0.97	1
70-20-10	Euclidian	1	1
70-20-10	Manhattan	1	1

Table 8. Results for the Iris Database with 10 antibodies with suppression

percentage split	distance	acc	val
60-20-20	Euclidian	0.96	0.97
60-20-20	Manhattan	0.97	0.97
70-20-10	Euclidian	1	1
70-20-10	Manhattan	1	1

Table 9. Results from other classifiers to iris database

classifier	acc
J48 (Decision Tree)	0.95
Multi-Layer Perceptron	0.97
Naive Bayes	0.95

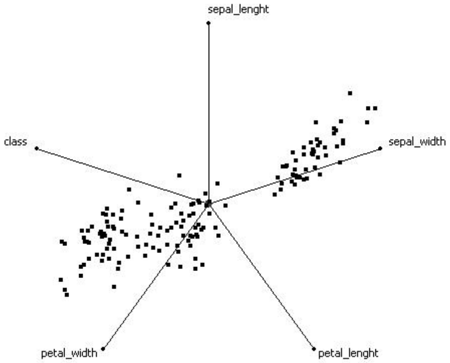


Fig. 7. The total amount of antibodies (training set) for the Iris database

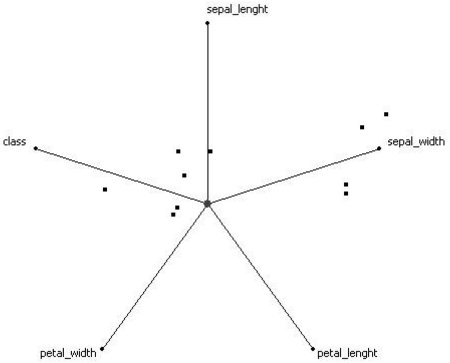


Fig. 8. Antibodies from Iris database after suppression and 60-40% split

4 Conclusions

This article proposed a new k-nearest-neighbor deterministic data classifier method using a suppression mechanism inspired on the behavior of biological immune systems. Particularly, the idea came from the way lymph nodes and T-helper cells behave towards lymphocytes no longer needed in the organism or those who self attack.

Basically, the predictor works with a database divided into three parts. The first one represents the antibodies and have the great proportion of the studied database, so that it can cover all the space. The second is the test set, responsible to help the system to eliminate those antibodies that were not used . The principle is, basically, those antibodies who recognized presented antigens remain on the population. The other ones are suppressed – taken away from the population. The third set is the validation one. Its role is to make sure that the suppression mechanism worked, retesting the system.

Experiments were made using benchmarks with two or more classes and the results were satisfactory. There was just a little proportion of mistakes between test and validation set. As next steps of this work, it would be necessary to further investigate multi-class problems and databases with unbalanced examples. Also, apply some hybrid techniques, such as fuzzy logic [30] or support vector machines [31] to try to reduce cross-reaction and improve performance.

References

1. Gordon, A.: Classification. Chapman and Hall, London (1981)
2. Espínola, R.P., Ebecken, N.F.F.: On extending f-measure and g-mean metrics to multi-class problems. DATA MINING VI - Data Mining, Text Mining and Their Business Applications 1, 25–34 (2005)
3. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco (2001)
4. Mitchell, T.: Machine Learning. McGraw-Hill, New York (1997)
5. Watkins, A.: AIRS: A resource limited immune classifier. Master's thesis, Mississipi State University (2001)
6. Watkins, A.B., Boggess, L.C.: A resource limited artificial immune classifier. In: Congress on Evolutionary Computation, IEEE World Congress on Computational Intelligence, Honolulu, HI, USA, pp. 926–931. IEEE Computer Society Press, Los Alamitos (2002)
7. Watkins, A.B., Boggess, L.C.: A new classifier based on resource limited artificial immune systems. In: Congress on Evolutionary Computation, IEEE World Congress on Computational Intelligence, Honolulu, HI, USA, pp. 1546–1551. IEEE Computer Society Press, Los Alamitos (2002)
8. Watkins, A., Timmis, J.: Artificial immune recognition system AIRS: Revisions and refinements. In: Timmis, J., Bentley, P. (eds.) 1st Intl. Conference on Artificial Immune Systems (ICARIS2002), pp. 173–181. University of Kent (2002)
9. Goodman, D., Boggess, L., Watkins, A.: An investigation into the source of power for airs, an artificial immune classification system. In: Intl. Joint Conference on Neural Networks (IJCNN'03), Portland, OR, USA, pp. 1678–1683 (2003)
10. Watkins, A., Timmis, J., Boggess, L.: In: Artificial Immune Recognition System (AIRS): An Immune-Inspired Supervised Learning Algorithm, vol. 5, pp. 291–317. Springer, Netherlands (2004)
11. Watkins, A., Timmis, J.: Exploiting 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)
12. Watkins, A.: Exploiting Immunological Metaphors in the Development of Serial, Parallel, and Distributed Learning Algorithms. PhD thesis, University of Kent, Canterbury, UK (2005)
13. Marwah, G., Boggess, L.: Artificial immune systems for classification: Some issues (2002)
14. Carter, J.H.: The immune system as a model for pattern recognition and classification. Journal of the American Medical Informatics Association 7, 28–41 (2000)
15. Timmis, J., Neal, M., Hunt, J.: An artificial immune system for data analysis. Biosystems 55, 143–150 (2000)
16. Timmis, J., Neal, M.: A resource limited artificial immune system for data analysis. Knowledge Based Systems 14, 121–130 (2001)

17. de Castro, L.N., Zuben, F.J.V.: Ainet: An artificial immune network for data analysis. In: Hussein, A., Abbass, R.A.S., Newton, C.S. (eds.) Data Mining: A Heuristic Approach. Idea Group Publishing, USA (2001)
18. Timmis, J., Neal, M.J.: A Resource Limited Artificial Immune System for Data Analysis. In: Timmis, J., Neal, M.J. (eds.) Research and Development in Intelligent Systems XVII. Proceedings of ES2000, Cambridge, UK, pp. 19–32 (2000)
19. Knight, T., Timmis, J.: Aine: An immunological approach to data mining. In: Cercone, N., Lin, T., Wu, X. (eds.) IEEE International Conference on Data Mining, San Jose, CA. USA, pp. 297–304. IEEE, New York (2001)
20. Timmis, J., Neal, M.: A resource limited artificial immune system for data analysis. Knowledge Based Systems 14, 121–130 (2001)
21. Knight, T., Timmis, J.: Assessing the performance of the resource limited artificial immune system AINE. Technical Report 3-01, Canterbury, Kent. CT2 7NF (2001)
22. Timmis, J., Knight, T.: Artificial immune systems: Using the immune system as inspiration for data mining. In: Abbass, H.A., Sarker, R.A., Newton, C.S. (eds.) Data Mining: A Heuristic Approach, pp. 209–230. Group Idea Publishing (2001)
23. de Castro, L.N.: Immune Engineering: Development of Computational Tools Inspired by the Artificial Immune Systems (in Portuguese). PhD thesis, DCA FEEC/UNICAMP, Campinas/SP, Brazil (August 1998 to May 2001)
24. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach, 1st edn. vol. 1 (2002)
25. Janeway, C.A., Travers, P., Walport, M., Shlomchik, M.: Immunobiologia: O sistema imune na saúde e na doença (in Portuguese), 5th edn. Artes Médicas (2001)
26. Perelson, A.S., Oster, G.F.: The shape space model. Journal of Theoretical Biology 81, 645–670 (1979)
27. Perelson, A.S., Weisbuch, G.: Immunology for physicists. Reviews of Modern Physics 69, 1219 (1997)
28. Voronoi, G.: Nouvelles applications des paramétres continus á la théorie des formes quadratiques. J. für die Reine und Angewandte Mathematik 133, 97–178 (1907)
29. Newman, D.J., Hettich, S., Merz, C.B.C.: UCI repository of machine learning databases (1998)
30. Ross, T.J.: Fuzzy Logic with Engineering Applications. McGraw Hill, New York (1995)
31. Cristianini, N., Shawe-Taylor, J.: An Introduction to support vector machines (and other kernel-based learning methods). Cambridge University Press, Cambridge (2000)

The Influence of Diversity in an Immune–Based Algorithm to Train MLP Networks

Rodrigo Pasti[1] and Leandro Nunes de Castro[2]

[1] State University of Campinas
[2] Catholic University of Santos

Abstract. This paper has three main goals: *i*) to employ an immune-based algorithm to train *multi-layer perceptron* (MLP) neural networks for pattern classification; *ii*) to combine the trained neural networks into ensembles of classifiers; and *iii*) to investigate the influence of diversity in the classification performance of individual and ensembles of classifiers. Two different classes of algorithms to train MLP are tested: bio-inspired, and gradient-based. Comparisons among all the training methods are presented in terms of classification accuracy and diversity of the solutions found.

Keywords: multi-layer perceptrons, bio-inpired algorithms, ensembles.

1 Introduction

There are currently a large number of computational techniques to solve data analysis and pattern recognition problems. Statistical and computational intelligence techniques are largely applied to these areas. Part of the efforts to design new computational algorithms to solve complex problems are based on the understanding of mechanisms, behaviors and dynamics of natural (biological, chemical and physical) systems, and implementations of methods inspired by these natural phenomena, systems, and processes. The research area that follows these principles has been called *bio-inspired* or *natural computing* [11]. Among the main natural computing approaches, *artificial immune systems* [9] have gained particular importance as a feasible alternative to solve complex problems.

The present paper investigates the use of bio-inspired algorithms to train MLPs, with particular emphasis on the application of an immune-based algorithm, named opt-aiNet [10], to adjust MLP network weights. Other algorithms, such as particle swarm optimization (PSO) [5], [6], an evolutionary algorithm (EA) [20], [21], [1] and the first-order backpropagation with momentum (BPM) [19] are also employed to the same task. The core issue to be discussed is how the diversity of the solutions found by the bio-inspired algorithms, which correspond to the weight sets determined for the MLP nets, influence the classification performance of the neural networks. To deepen the investigation, the neural classifiers trained are combined into ensembles so as to provide a better assessment of the influence of diversity of individual classifiers when combined.

A number of classification problems are used to evaluate the performance of the algorithms, which is measured by their classification accuracy and the diversity of the

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 71–82, 2007.

classifiers generated. The results show that most bio-inspired algorithms generate sets of very similar solutions that do not contribute significantly to an increase in performance of the ensembles of classifiers. The results of opt-aiNet are, by contrast, naturally more diverse, thus leading to improved performance of the ensembles.

2 Algorithms Used to Train MLPs

MLP network training can be viewed as a function approximation problem in which the network parameters (weights and biases) are adjusted, during training, in an effort to minimize (optimize) an error function between the network output and the desired output. Thus, virtually any search and optimization algorithm can be used to train this type of neural network. The error function utilized in the present work is the Mean Square Error (MSE):

$$MSE = \frac{1}{o}\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{o}\delta_{ij}^{2}(t)\,, \tag{1}$$

where $\delta_{ij}(t)$ is the error for output neuron i and input pattern j at time instant t; o is the number of network outputs, and N is the number of input patterns.

2.1 opt-aiNet

The term opt-aiNet stands for *optimization version of an artificial immune network* and corresponds to an adaptation procedure to perform search and optimization based on some immune principles [10]. It emphasizes the clonal selection and affinity maturation principles, together with the immune network theory [9].

Some features and metaphors of opt-aiNet:

- It is a multi agent system in which the individuals of the population are the immune cells that recognize the antigens (candidate solutions);
- Each cell corresponds to a real-valued vector in an Euclidean space;
- The affinity between two cells or individuals is quantified using their Euclidean distance;
- Clones are identical copies of the parent cell, but all clones are subjected to a controlled mutation that corresponds to a perturbation in their attributes;
- Selection is performed within each clone and including the parent cell;
- The fitness of each cell is the value of the objective function when evaluated for the cell; and
- Network stimulation is not accounted for, but suppression is mimicked by removing all the worst cells among those whose Euclidean distance is smaller than a pre-defined threshold σ_s.

The controlled mutation takes into account the fitness of the parent cell that generated the clones: the higher the fitness, the smaller the mutation rate, and vice-versa, as follows:

$$\mathbf{c}' = \mathbf{c} + \alpha\, N(0,1) \tag{2}$$
$$\alpha = (1/\beta)\exp(-f^{*})$$

where **c** is the vector that corresponds to the clone, **c'** is the mutated clone, f^* is the fitness of the parent cell normalized in the interval [0,1].

The opt-aiNet algorithm is summarized below.

```
Algorithm = opt-aiNet(β,σs,d)
  . Randomly initialize a number of cells taking into account
  the variables' domains
  while population-not-stabilized
       while average-fitness-does-alter
               . Generate a fixed number of clones for each cell
               . Mutate the clones following Eq. (1)
               . Determine the fitness of each clone
               . Select among the clones
               . Determine the average fitness
       End while
       . Determine the affinity among all cells
       . Suppress cells according to the threshold σs
       . Introduce new cells randomly generated. The number of
         new cells is d% the total number of cells.
  End while
End algorithm
```

Algorithm 1. Code for the opt-aiNet algorithm

Roughly, the computational cost of the algorithm is quadratic in relation to the number n of cells in the network at a given time instant: $O(n^2)$. The convergence criterion of the outer loop is based on a stabilization of the number of cells in the network after suppression; that is, if the number of cells does not change from one iteration to the other, then the network has converged. From a behavioral perspective, a distinguishing feature of opt-aiNet is its capability of locating and maintaining a memory of multiple optima solutions.

In order to apply opt-aiNet to train multi-layer perceptron neural networks, few modifications have to be introduced in the algorithm. As the MLP error surface is a function of the weights and biases, training the network becomes the problem of searching for the most suitable weight set that minimizes the error. Therefore, a network cell will correspond to the weight vector of the whole network, including the bias terms for the hidden and output neurons. The objective function of opt-aiNet is the MLP error. Assuming networks with a fixed, user-defined, number of hidden units in a single hidden layer, the errors that remain to be evaluated are the optimization and the generalization errors. These errors can be estimated, for instance, by using cross-validation [14], [15]. The network error for the training set is the optimization error and the error for the test set estimates the generalization error.

For the use of opt-aiNet to train MLPs three modifications were introduced. In the original opt-aiNet algorithm, when the number of cells in the network stabilizes and there is no significant change in the average network fitness, evolution stops. These criteria allow opt-aiNet to suppress useless cells (those that do not recognize any input pattern), include new cells in the network and keep searching. The first modification is related to the stopping criterion, which is now a fixed number of iterations.

Mutation

Preliminary tests with the use of opt-aiNet to train multi-layer perceptrons suggested that parameter β (Eq. (2)) exert an important influence on the behavior of the algorithm. It was noted that values (much) less than 10 promote a large perturbation in the candidate solutions and, consequently, the saturation of the sigmoidal MLP network neurons. When β is approximately 10, however, the algorithm converges more quickly; that is, the weight adjustment process stabilizes faster, but is more subject to local optima solutions due to premature convergence. Greater values of β promote a finer adjustment in the weights, because they will result in smaller step sizes, but at the expense of a higher computational cost. It was noted that β = 100, for instance, is a value that results in a sufficiently fine adjustment in the weights. Based on these observations, a simple heuristics is proposed here to adjust the value of β during evolution:

- $0 \leq 1-f^* < 0{,}2 \rightarrow \beta = 100$
- $0{,}2 \leq 1-f^* < 0{,}4 \rightarrow \beta = 75$
- $0{,}4 \leq 1-f^* < 0{,}6 \rightarrow \beta = 50$
- $0{,}6 \leq 1-f^* < 0{,}8 \rightarrow \beta = 25$
- $0{,}8 \leq 1-f^* \leq 1 \rightarrow \beta = 10$

where f^* is the normalized fitness. Note that, in this heuristics, the higher the fitness value, the smaller the perturbation in Eq. (1).

Suppression

The preliminary tests performed with opt-aiNet to train MLPs also showed that the suppression mechanism depends on the dimension of the problem to be solved, in this case the number of weights and biases to be adjusted. This is because suppression in opt-aiNet considers the Euclidean distance among solutions and, for non-normalized vectors, the greater the number of dimensions, the larger the distance is allowed to be. As the problems to be studied here require varying neural network dimensions, it is relevant to make suppression independent of this parameter. Thus, the following steps are proposed to perform suppression:

- The weight vectors are normalized in the [0,1] interval;
- The maximal distance between the normalized solutions is determined;
- The distance matrix among solutions is divided by the maximal distance.

Parameter σ_s is now determined as proportional to the normalized distances, for instance, 0.1 times the normalized distance.

2.2 Other Algorithms to MLP Training

In this paper two bio-inspired algorithms, namely the standard particle swarm optimization (PSO) algorithm and an evolutionary algorithm, were implemented for performance comparisons with opt-aiNet in the training of MLP neural networks. These will be briefly described in the following.

The particle swarm optimization algorithm, based on sociocognition, has been largely used for solving search and optimization problems [6]. In PSO each individual is a candidate solution (weight set for the MLP) that has an associated performance index that allows it to be compared with the other particles. The version utilized here has global neighborhood and the inertia weight term [6].

The evolutionary algorithm used here combines ideas from genetic algorithms [1], [20], [21] and evolution strategies [1], [20], [21]. Operators typical of genetic algorithms, such as uniform recombination, point mutation and roulette wheel selection, are used, together with the self-adaptation of evolution strategies. The encoding scheme employed was floating point numbers due to the nature of the task to be performed – train MLP neural networks. An elitist strategy is used to save the best individual in the population. Specifically, the following operators used were:

- Uniform crossover [1];
- Tournament selection; and
- Self-adaptive Gaussian mutation [20], following Eq. (3):

$$\sigma_i^{t+1} = \sigma_i^{t} \exp(\tau' . N(0,1) + \tau . N_i(0,1)) \qquad (3)$$
$$\mathbf{w}_i^{t+1} = \mathbf{w}_i^{t} + \sigma_i^{t+1} . N_i(0,1)$$

The last algorithm utilized here for comparison is a first-order method. Each minimization step is given in the opposite direction of the gradient vector and is controlled by the learning rate α. An additional term (β) usually avoids oscillation in the error behavior, because it can be interpreted as the inclusion of an approximation of a second-order information [19]. The resulting algorithm is called the standard *back-propagation with momentum* (BPM).

3 Ensembles of Classifiers

Ensembles are aimed at using a finite collection of independent components (classifiers or regressors), such that a single response is provided to the problem by the whole ensemble. The application of this idea to neural networks was introduced by [12], when they showed that the generalization capability can be improved when the outputs of various networks are combined after individual trainings. Building an ensemble involves, basically, three steps: 1- *Component generation*, 2- *Component selection,* 3- *Combination of components*.

Although the ensemble may present substantial gains in performance when compared with the best component in isolation, its performance may still be unsatisfactory [7]. Every ensemble component must present a good individual performance and they must be uncorrelated with each other. The correlation among components is related to their diversity; the higher the diversity, the smaller their correlation. Two components (classifiers) are uncorrelated if they present different classification errors for the same data. Techniques for pre-processing data, such as *boosting* and *bagging* will not be used [22], [8] and the component generation step will be restricted to MLP training because one of the main goals of this work is to investigate the original diversity generated by each algorithm.

3.1 Component Selection

Two component selection methods were used in this work: *i*) *constructive selection*; and *ii*) *pruning selection*. In the constructive selection method, a non-decreasing ordering of the components is made based on the classification error for the test set. The component with the smallest error is inserted into the ensemble. In the sequence, the second-best component in inserted, and, if the ensemble performance is improved, then this component is maintained, else it is removed from the ensemble. The process is repeated until all the components have been evaluated. In the selection by pruning, a non-increasing ordering of the components is made based on the classification error for the test set, and the ensemble starts will all classifiers included. Then, the first classifier (the one with worst individual performance) is removed from the ensemble, and, if the ensemble performance improves, then it is kept out of the ensemble, else it is maintained in the ensemble. This process is repeated until all classifiers have been evaluated.

3.2 Component Combination

Three methods to combine classifiers were tested in this work: *i*) *voting*; *ii*) *simple average*; and *iii*) *optimal linear combination*. In the voting method, the result supported by most components is defined as the ensemble output. Although this method has a low computational cost, if there are many components performing poorly, then the ensemble output may be negatively influenced. The simple average, as the name implies, determines the ensemble output $y_e(\cdot)$ by taking the average of all outputs provided by the individual classifiers $y_i(\cdot)$, $i = 1,\dots,p$, where p is the number of classifiers. The optimal linear combination consists of weighting the outputs of all components based on their individual performances assuming that a linear combination of components is used [17], [18].

3.3 Diversity Measures

The diversity measures aim at evaluating the decorrelation level among the classifiers, and most of them are derived from statistics. The most common diversity measure consists of using the *oracle* output of the classifier; that is, given a certain input pattern, the oracle output simply says if the classifier presented a correct classification or not. In this work two diversity measures were investigated [13]: *i*) *disagreement measure*; and *ii*) *entropy measure*.

Disagreement Measure

This measure evaluates the classifiers pairwise and then takes the average of the disagreement values. It is the ratio between the number of observations in which one classifier is correct and the other is incorrect to the total number of observations. For two classifiers C_i and C_k:

$$Dis_{i,k} = \frac{N^{01} + N^{10}}{N^{11} + N^{10} + N^{01} + N^{00}} \tag{4}$$

where N^{ab} is the number of elements $\mathbf{y}_j \in Y$ in which $y_{j,i} = a$ and $y_{j,k} = b$, according with Table 1

Table 1. Relationship between a pair of classifiers

	C_k correct (1)	C_k incorrect (0)
C_i correct (1)	N^{11}	N^{10}
C_i incorrect (0)	N^{01}	N^{00}

For the set of all L classifiers, the total diversity is:

$$Div = \frac{2}{L(L-1)} \sum_{i=1}^{L-1} \sum_{k=i+1}^{L} Dis_{i,k} \tag{5}$$

Entropy Measure

The highest diversity among classifiers for a particular sample $\mathbf{x}_i \in X$ corresponds to $\lfloor L/2 \rfloor$ of the votes in X with the same value (0 or 1) and the other $L - \lfloor L/2 \rfloor$ with the alternative value. If all classifiers are 0 or 1, then there is no disagreement among them and, consequently, no diversity. Let us denote $l(\mathbf{x}_j)$ the number of classifiers that correctly classify $\mathbf{x}_j$:

$$l(\mathbf{x}_j) = \sum_{i=1}^{L} y_{j,i} \tag{6}$$

A possible entropy measure is

$$E = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{(L - \lceil L/2 \rceil)} \min[l(\mathbf{x}_j), L - l(\mathbf{x}_j)] \tag{7}$$

which varies over the interval [0,1], being 0 the smallest diversity and 1 the largest.

4 Performance Evaluation

To assess the performance of each algorithm in the process of MLP weight adjustment, four data sets from the repository of machine learning data sets of the University of California at Irvine [3] were chosen.

4.1 Materials and Methods

The k-fold cross-validation method [14], [15], [19] with $k = 10$ was used in the experiments to be reported here. It consists of dividing the data set into k folders, training the network with the data from $k - 1$ folders and testing it with the data from the other folder not used for training. After that, the average classification (optimization and generalization) errors are calculated.

The population-based approaches (EA, PSO and opt-aiNet) all provide a number n of distinct solutions (individuals). Each of the n solutions is tested with the k folders and those that present the best performance are taken and combining in an ensemble.

For the BPM algorithm, a number n of trainings, thus solutions, was performed such that an even number of solutions is available for all methods. In the particular case of opt-aiNet, it automatically defines the number of cells in the network and, thus, to perform the comparisons, opt-aiNet was run first and the resultant number of cells was used for all other algorithms. The following ensembles were investigated: ***Best***: the best solution among the n generated ones is selected (without combining the classifiers). ***Ensemble* 1 (WS+V)**: *without selection* plus *voting*. ***Ensemble* 2 (WS+SA)**: *without selection* plus *simple average*. ***Ensemble 3* (SS+OLC)**: *without selection* plus *optimal linear combination*. ***Ensemble 4* (C+V)**: *constructive* method plus *voting*. ***Ensemble 5* (C+MS)**: *constructive* method plus *simple average*. ***Ensemble 6* (C+OLC)**: *constructive* method plus *optimal linear combination*. ***Ensemble 7* (P+V)**: *pruning* method plus *voting*. ***Ensemble 8* (P+MS)**: *pruning* method plus *simple average*. ***Ensemble 9* (P+OLC)**: *pruning* method plus *optimal linear combination*.

The attributes of all data were normalized within the [−1, 1] interval due to the use of MLP with hyperbolic tangent in the hidden units and with linear outputs. The data sets and network architectures [n. inputs, n. of hidden units, n. of outputs] used for evaluation can be summarized as follows: **Wine:** 178 samples divided into three classes, and with 13 attributes. Architecture: [13,10,2]. **Wisconsin Breast Cancer:** 699 samples divided into two classes, and with 9 attributes. Network architecture: [9,10,1]. **Ionosphere:** 351 samples divided into two classes, and with 34 attributes. Architecture: [34,10,1]. The parameters used for all algorithms are described in Table 2.

Table 2. Training parameters for the algorithms evaluated

PSO	opt-aiNet	EA	BPM
φ_1 : 2.05	d: 50%	$p_r = 0.5$	$\alpha = 0.0001$
φ_2 : 2.05	$\sigma_s = 20$*	Max-it = 10^4	$\beta = 0.5$
k: 0.729	N: 10		Max-it = 10^4
$v_{\max}$: 2	N_c: 10		
N_p: 10	Max-it : 10^4		
Max-it: 10^4			

* For the Wine dataset $\sigma_s = 30$.

To evaluate the statistical significance of the difference between the averages of all errors the t-test [4] was employed pairwise. It is assumed that the samples are independent and normally distributed. The independence of the samples can be verified due to the random initialization in each algorithm, and normality could be observed in preliminary experiments. The following analyses and comparisons will be made:

- Opt-aiNet will be compared pairwise with all other techniques used. The averages that **do not** present a statistically significant difference (to a 5% level) will be marked in **bold**.

- Comparisons of the average of the best solutions and the ensembles. If the ensemble results in a smaller classification error that is statistically less than the best average, ***best***, then this value will be detached with a '*'.
- The best solutions of all are shadowed.

4.2 Experimental Results

Wine Dataset

Table 3 shows that opt-aiNet obtained results significantly better than the other algorithms in terms of the best solution found and ensemble generated. In some cases the results of BPM had no statistical difference to those of opt-aiNet. It is also interesting to note that, sometimes, the use of ensembles led to a worst performance than the best solution, which is due to the presence of very poor solutions in isolation. The diversity of each algorithm is presented in Table 4.

Table 3. Classification error for the test set of all algorithms for the Wine dataset. **Best**: best solution found by each method; **WS**: without selection; **C**: constructive method; **P**: pruning method; **V**: voting, **SA**: simple average; **OLC**: optimal linear combination; **bold**: indicates the absence of statistical different in relation to opt-aiNet; **star (*)**: significant improvement with the use of ensembles; and shadow: best solution of all.

	opt-aiNet	**PSO**	**AE**	**BPM**
Best	0.11 ±0.23	2.72 ±1.07	2.02 ±0.73	1.89 ±3.25
WS + V	1.54 ±0.55	3.27 ±1.25	2.58 ±1.14	41.33 ±8.36
WS + SA	1.74 ±0.26	3.44 ±1.40	2.63 ±1.10	23.43 ±6.77
WS + OLC	1.15 ±0.48	3.27 ±1.35	2.58 ±1.08	2.89 ±3.32
C + V	0.11 ±0.23	2.72 ±1.07	2.02 ±0.73	1.89 ±3.25
C + SA	0.11 ±0.23	2.72 ±1.07	2.02 ±0.73	1.89 ±3.25
C + OLC	0.11 ±0.23	1.99 ±0.85	1.63 ±0.74	1.72 ±3.33
P + V	1.31 ±0.48	3.27 ±1.25	2.58 ±1.14	39.01 ±6.75
P + SA	1.51 ±0.39	3.33 ±1.29	2.52 ±1.11	21.15 ±7.39
P + OLC	0.99 ±0.55	3.27 ±1.35	2.52 ±1.11	2.65 ±3.19

Table 4. Diversity of solutions for the Wine dataset

	opt-aiNet	**PSO**	**AE**	**BPM**
Entropy measure	0.0581	0.0072	0.0043	0.6093
Disagreement measure	0.0469	0.0050	0.0034	0.4030

Ionosphere Dataset

Table 5 shows that opt-aiNet performed better in almost all cases, and almost all algorithms improved their performances when their solutions were combined into ensembles, with the exception of the evolutionary algorithm. The opt-aiNet showed the greatest diversity (Table 6).

Table 5. Classification error for the test set of all algorithms for the Ionosphere dataset. **Best**: best solution found by each method; **WS**: without selection; **C**: constructive method; **P**: pruning method; **V**: voting, **SA**: simple average; **OLC**: optimal linear combination; **bold**: indicates the absence of statistical different in relation to opt-aiNet; **star (*)**: significant improvement with the use of ensembles; and shadow: best solution of all.

	opt-aiNet	PSO	AE	BPM
Best	4.50 ±0.38	11.59 ±0.94	10.57 ±0.84	10.97 ±0.24
WS + V	9.19 ±0.63	13.42 ±1.33	11.19 ±1.10	12.39 ±0.24
WS + SA	9.48 ±0.59	13.56 ±1.32	11.19 ±1.10	12.36 ±0.15
WS + OLC	0.17 ±0.46*	11.20 ±2.10	11.19 ±1.10	12.81 ±0.23
C + V	4.19 ±0.39	11.59 ±0.94	10.57 ±0.84	10.83 ±0.30
C + SA	4.42 ±0.39	11.56 ±0.95	10.57 ±0.84	10.97 ±0.24
C + OLC	4.39 ±0.41	10.83 ±0.88	10.05 ±1.07	10.63 ±0.33*
P + V	8.60 ±0.78	13.28 ±1.21	11.10 ±1.02	12.33 ±0.23
P + SA	8.91 ±0.66	13.48 ±1.29	11.16 ±1.08	12.30 ±0.18
P + OLC	0.15 ±0.37*	9.92 ±2.17*	11.16 ±1.08	12.50 ±0.21

Table 6. Diversity of solutions for the Ionosphere dataset

	opt-aiNet	PSO	AE	BPM
Entropy measure	0.1489	0.0247	0.0053	0.0377
Disagreement measure	0.1082	0.0182	0.0039	0.0311

Breast Cancer Dataset

Table 7 present the best isolated performances opt-aiNet. The use of ensembles led to improvements for BPM, but for no bio-inspired training algorithm. For this problem the best results were determined by the opt-aiNet approach, which also presented the highest diversity (Table 8).

Table 7. Classification error for the test set of all algorithms for the Breast Cancer dataset. **Best**: best solution found by each method; **WS**: without selection; **C**: constructive method; **P**: pruning method; **V**: voting, **SA**: simple average; **OLC**: optimal linear combination; **bold**: indicates the absence of statistical different in relation to opt-aiNet; **star (*)**: significant improvement with the use of ensembles; and shadow: best solution of all.

	opt-aiNet	PSO	AE	BPM
Best	2.56 ±0.17	2.88 ±0.13	2.98 ±0.27	3.01 ±0.00
WS + V	3.11 ±0.07	3.42 ±0.14	**3.09 ±0.30**	3.18 ±0.06
WS + SA	3.12 ±0.06	3.39 ±0.17	**3.09 ±0.30**	3.25 ±0.12
WS + OLC	2.72 ±0.00	3.35 ±0.15	3.09 ±0.30	3.21 ±0.07
C + V	2.53 ±0.14	2.88 ±0.13	2.98 ±0.27	3.01 ±0.00
C + SA	2.56 ±0.17	2.88 ±0.13	2.98 ±0.27	3.01 ±0.00
C + OLC	2.53 ±0.15	2.88 ±0.13	2.95 ±0.25	2.92 ±0.10*
P + V	3.11 ±0.07	3.40 ±0.15	**3.08 ±0.28**	3.18 ±0.06
P + SA	3.12 ±0.06	3.38 ±0.15	**3.09 ±0.30**	3.22 ±0.12
P + OLC	2.72 ±0.00	3.30 ±0.17	3.09 ±0.30	3.19 ±0.07

Table 8. Diversity of solutions for the Breast Cancer dataset

	opt-aiNet	**PSO**	**AE**	**BPM**
Entropy measure	0.1052	0.0208	0.0178	0.1449
Disagreement measure	0.0783	0.0153	0.0128	0.1021

5 Discussion

By focusing on the use of bio-inspired and gradient-based algorithms to train multilayer perceptron neural networks it is possible to remark:

- The solutions found by opt-aiNet were in general better than those of the other bio-inspired algorithms. This is observed in the isolated (best) solutions and in the ensembles of MLP classifiers trained by opt-aiNet as well.
- The quality of the ensembles may be associated with the diversity of the components, for almost invariably more diverse solutions led to improved ensembles when compared with the isolated best solutions.
- There seems not to be an ensemble method (selection plus combination) that is more adequate to all classification problems. However, the techniques WS+OLC (without selection + optimal linear combination) achieved good results in some cases for many algorithms.
- The results obtained by the diversity measures indicate that, as expected, decorrelated classifiers tend to result in improved ensembles. In the particular case of PSO and the EA implemented, the ensembles performed poorly, probably because most of the individuals generated by these methods tend to cluster around the best solution found, thus compromising the diversity.

References

[1] Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Heidelberg

[2] Rumelhart, D.E., McClelland, J.L.: The PDP Research Group, Parallel Distributed Processing: Explorations in the Microstructure of Cognition. In: Foundations, vol. 1, The MIT Press, Cambridge (1986)

[3] Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases. University of California, Department of Information and Computer Science, Irvine, CA (1998), [http://www.ics.uci.edu/ mlearn/MLRepository.html]

[4] Witten, I.H., Frank, E.: Data Mining: Practical Learning Tool and Techniques, 2nd edn. Morgan Kauffman Publishers, San Francisco (2005)

[5] Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proc. of the IEEE Int. Conf. on Neural Networks. Perth, Australia, vol. 4, pp. 1942–1948 (1995)

[6] Kennedy, J.: Particle Swarm: Optimization Bases on Sociocognition. In: de Castro, L.N., Von Zuben, F.J. (eds.) Capítulo do livro Recent Developments in Biologically Inspired Computing, pp. 235–268. Idea Group Publishing, Capítulo X (2004)

[7] Tumer, K., Ghosh, J.: Error correlation and error reduction in ensemble classifiers. Connection Science, Special Issue on Combining Artificial Neural: Ensemble Approaches 8(3&4), 385–404 (1996)

[8] Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)

[9] de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, Heidelberg (2002)

[10] de Castro, L.N., Timmis, J.: An Artificial Immune Network for Multimodal Function Optimization. Proc. of the IEEE Congress on Evolutionary Computation 1, 674–699 (2002)

[11] De Castro, L.N.: Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications. CRC Press, Boca Raton, USA (2006)

[12] Hansen, L.K., Salamon, P.: Neural Network Ensemble. IEEE Transactions On Pattern Analysis And Machine Intelligence 12, 10 (1996)

[13] Kuncheva, L.I., Whitaker, C.J.: Measure of Diversity in Ensembles an Their Relationship with Ensemble Accuracy. Machine Learning 51, 181–207 (2003)

[14] Prechelt, L.: Automatic Early Stopping Using Cross Validation: Quantifying the Criteria. Neural Networks 11(4), 761–767 (1998)

[15] Prechelt, L.: Early Stopping – but when? Technical Report (1997)

[16] Pasti, R., de Castro, L.N.: An Immune and a Gradient-Based Method to Train Multi-Layer Perceptron Neural Networks. In: Proc. International Joint Conference on Neural Networks (World Congress of Computational Intelligence), pp. 4077–4084 (2006)

[17] Hashem, S., Schmeiser, B.: Improving Model Accuracy Using Optimal Linear Combinations of Trained Neural Networks. IEEE Transactions on Neural Networks 6, 3 (1995)

[18] Hashem, S.: Optimal Linear Combinations of Neural Networks. Neural Networks 10(4), 599–614 (1997)

[19] Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice-Hall, Englewood Cliffs (1999)

[20] Bäck, T., Fogel, D.B., Michalewicz, Z.: Evolutionary Computation 1 Basic Algorithms and Operator. Institute of Physiscs Publishing (IOP), Bristol and Philadelphia (2000)

[21] Bäck, T., Fogel, D.B., Michalewicz, Z.: Evolutionary Computation 2 Advanced Algorithms apnd Operators. Institute of Physiscs Publishing (IOP), Bristol and Philadelphia (2000)

[22] Freund, Y., Shapire, R.: Experiments with new boosting algorithm. In: Proc. of the 13th International Conference on Machine Learning, pp. 149–156 (1996)

Applying Biclustering to Text Mining: An Immune-Inspired Approach

Pablo A.D. de Castro, Fabrício O. de França, Hamilton M. Ferreira,
and Fernando J. Von Zuben

Laboratory of Bioinformatics and Bio-Inspired Computing - LBIC
School of Electrical and Computer Engineer – FEEC
University of Campinas – UNICAMP
Campinas-SP, Brazil
{pablo,olivetti,hmf,vonzuben}@dca.fee.unicamp.br

Abstract. With the rapid development of information technology, computers are proving to be a fundamental tool for the organization and classification of electronic texts, given the huge amount of available information. The existent methodologies for text mining apply standard clustering algorithms to group similar texts. However, these algorithms generally take into account only the global similarities between the texts and assign each one to only one cluster, limiting the amount of information that can be extracted from the texts. An alternative proposal capable of solving these drawbacks is the biclustering technique. The biclustering is able to perform clustering of rows and columns simultaneously, allowing a more comprehensive analysis of the texts. The main contribution of this paper is the development of an immune-inspired biclustering algorithm to carry out text mining, denoted BIC-aiNet. BIC-aiNet interprets the biclustering problem as several two-way bipartition problems, instead of considering a single two-way permutation framework. The experimental results indicate that our proposal is able to group similar texts efficiently and extract implicit useful information from groups of texts.

Keywords: Artificial Immune System, Biclustering, Two-way Bipartition, Text mining.

1 Introduction

With the popularization of the web and the collaboration of users to produce digital contents, there was an expressive increase in the amount of documents in electronic format. Therefore, several text mining tools have been proposed to organize and classify these documents automatically, since a personal manipulation is becoming more and more prohibitive.

The main difficulty associated with automated analysis of documents is that textual information is highly subjective. Although there are many efficient data mining tools available in the literature, converting such information into a rather objective and computer-interpretable codification is far from being straightforward. Invariably, this scenario imposes several limitations to the performance of the analytical tools. On the other hand, while this conversion step is still not entirely satisfactory, there must be

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 83–94, 2007.

an effort to enhance the clustering and classification techniques in order to handle the noisy and incomplete information generated by methods for semantic interpretation.

Most of the text mining tools reported in the literature represent the documents by a vector of attributes [6]. After submitting the text to a filter capable of extracting irrelevant words and particles of composed words (filters may differ a lot and may present very distinct degrees of sophistication), each position of the vector is related to a word or textual expression of the preprocessed text, and each entry of the vector contains a number which says how many times the corresponding word or expression appears in the text. Therefore, the text mining tools performs several types of data analysis based on statistical properties presented by the attributes extracted from preprocessed texts.

In the context of text clustering, although standard clustering algorithms such as *k*-means, Self Organized Maps, and Hierarchical Clustering have been successfully applied to text mining, they present a well- known limitation when dealing with large and heterogeneous datasets: since they group texts based on global similarities in their attributes, partial matching cannot be detected. For instance, if two or more texts share only a subset of similar attributes, the standard clustering algorithms fail to identify this specificity in a proper manner. Besides, they assign a text to only one category, even when the texts are involved in more than one category.

A recent proposal to avoid these drawbacks is the so-called biclustering technique, which performs clustering of rows and columns simultaneously, allowing the extraction of additional information from the dataset [2]. Biclustering may be implemented considering a two-way permutation problem or performing several two-way bipartitions of the whole dataset, as will be clarified in Section 2.

The objective of this paper is twofold. The first one is to apply a biclustering technique to the text mining problem, centered on the two-way bipartition framework. To do that, a very flexible methodology is considered to implement multiple two-way bipartitions, characterized by the possibility of discarding an arbitrary number of rows (texts) and columns (attributes of the corresponding texts) of the original matrix. This methodology is in contrast with the ones that are capable of finding all biclusters in a matrix [12]. So, if we have a large matrix, the computational cost to generate the biclusters using the latter approach becomes prohibitive.

Aiming at proposing a feasible solution, the second objective is to present a novel heuristic-based methodology to generate the multiple two-way bipartitions. Once the generation of biclusters can then be viewed as a multimodal combinatorial optimization process, we will explore the already attested ability of immune-inspired algorithms to deal with this challenging scenario.

We evaluate the proposed methodology by applying it to a dataset which contains 60 texts classified into 3 different categories. The results indicate that our algorithm is able to group similar texts correctly. Besides, the biclusters present the more relevant words to represent a certain category. This information is useful when composing an intelligent search engine for guiding a user on its search for related documents on different areas.

The paper is organized as follows. Section 2 presents a brief introduction to the biclustering technique and its applications. Section 3 describes in details the proposed approach to generate biclusters. In Section 4, the experimental results are presented and analyzed. Finally, in Section 5 we conclude the paper and provide directions for further steps in the research.

2 A Brief Overview of Biclustering

In data mining, biclustering is referred to the process of simultaneously find clusters on the rows and columns of a matrix [2]. This matrix may represent different kinds of numerical data, such as objects and its attributes (comprising the rows and columns of the matrix, respectively).

There are several approaches to deal with the biclustering problem [8][10][11][13][14]. The most traditional one is to interpret biclustering as a two-way permutation problem, so that the purpose is to simultaneously reorder rows and columns of the original matrix, in an interactive manner, toward the production of multiple clusters in different regions of the obtained matrix, as illustrated in Fig. 1.

$$\begin{bmatrix} 1 & 2 & 3 & 1 \\ 3 & 3 & 2 & 2 \\ 1 & 2 & 2 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 2 & 3 & 1 \\ 1 & 2 & 2 & 1 \\ 3 & 3 & 2 & 2 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 & 3 & 2 \\ 1 & 1 & 2 & 2 \\ 3 & 2 & 2 & 3 \end{bmatrix}$$

Fig. 1. The original matrix is reordered (rows and columns) to generate the biclusters

Another possibility, which will be explored in this paper, is to create several sub-matrices from the original matrix aiming at maximizing some index designed to measure similarity, or alternative clustering aspects, in these sub-matrices. As the construction of the sub-matrices involves defining which rows and columns of the original matrix will be included and which ones will be excluded, we may interpret the biclustering as multiple two-way partition problems, as illustrated in Fig. 2.

$$\begin{bmatrix} 1 & 2 & 3 & 1 \\ 3 & 3 & 2 & 2 \\ 1 & 2 & 2 & 1 \end{bmatrix} \Rightarrow \begin{array}{l} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{array}{l} rows = \{1,3\} \\ cols = \{1,4\} \end{array} \\ \begin{bmatrix} 2 & 2 \\ 2 & 1 \end{bmatrix} \begin{array}{l} rows = \{2,3\} \\ cols = \{3,4\} \end{array} \end{array}$$

Fig. 2. Two biclusters extracted from the original matrix

Additional aspects may be considered to distinguish biclustering techniques. They can be classified by: (*i*) the type of biclusters they find; (*ii*) the structure of these biclusters; and (*iii*) the way the biclusters are discovered.

The type of the biclusters is related to the concept of similarity between the elements of the matrix. For instance, some algorithms search for constant value biclusters, while others search for coherent values of the elements or even for coherent evolution biclusters.

The structure of the biclusters can be of many types. There are single bicluster algorithms, which find only one bicluster in the center of the matrix; the exclusive columns and/or rows, in which the biclusters cannot overlap in either columns or rows of the matrix; arbitrary positioned, overlapping biclusters and overlapping biclusters with hierarchical structure.

The way the biclusters are discovered refers to the number of biclusters discovered per run. Some algorithms find only one bicluster, others simultaneously find several biclusters and some of them find small sets of biclusters at each run.

Besides, there are nondeterministic and deterministic algorithms. Nondeterministic algorithms are able to find different solutions for the same problem at each execution, while deterministic ones produce always the same solution. The algorithm used in this paper is nondeterministic.

The biclustering approach covers a wide scope of different applications. The main motivation is to find data points that are correlated under only a subset of the attributes. Usual clustering methods cannot identify this type of local correlation. Some examples of biclustering applications are dimensionality reduction [1], information retrieval and text mining [5], electoral data analysis [9], and biological data analysis [1].

3 An Immune-Inspired Algorithm for Biclustering

Since the generation of biclusters can be viewed as a multimodal combinatorial optimization problem, we have considered an immune-inspired algorithm to generate them. The first attempt to synthesize an immune-inspired method for biclustering has taken copt-aiNet (Artificial Immune Network for Combinatorial Optimization) as the search engine [4]. Our approach is another extension of the aiNet algorithm, which was proposed by de Castro and Von Zuben [3], to deal with biclustering and is denoted BIC-aiNet, an artificial immune network for biclustering. Though copt-aiNet and BIC-aiNet are both derived from aiNet to deal with combinatorial optimization, BIC-aiNet considers the biclustering problem as multiple two-way bipartitions, and copt-aiNet has adopted the two-way permutation framework.

When compared with alternative approaches, immune-inspired algorithms present distinguishing characteristics that proved to be specially interesting for solving multimodal problems, such as: (*i*) the multi-population paradigm, where several populations are evolved at the same time (in contrast with usual population algorithms, where there is just one population that converges to only one solution); (*ii*) the dynamic control of the population size, which allows the automatic definition of the number of candidate solutions based on the characteristics of the problem; and (*iii*) the diversity maintenance property, which enhances the searching ability of the algorithm by allowing the coexistence of multiple alternative solutions in the same run.

Next, we present how the biclusters are represented in BIC-aiNet as well as its functioning.

3.1 Coding

Given a data matrix with n rows and m columns, the structure chosen to represent a bicluster (an individual in the population) on the algorithm for this data matrix is by using two ordered vectors. One vector represents the rows and the other one represents the columns, with length *n*' < *n* and *m*' < *m*, respectively. Each element in the vectors is an integer representing the index of row or column that is present at the bicluster. This representation tends to be more efficient than the binary representation (two vectors of length *n* and *m,* where the number 1 represents the presence of a given row/column and 0 the absence) once the biclusters are usually much smaller than the original data matrix.

Each individual in the population represents a single bicluster and may have a distinct value for *n*' and *m*'. Figure 3 shows an example of coding

$$\begin{bmatrix}1&2&3&1\\2&3&2&2\\1&2&2&1\end{bmatrix}$$

(a)

row:	**2**	**3**	
col:	**1**	**3**	**4**

(b)

$$\begin{bmatrix}2&2&2\\1&2&1\end{bmatrix}$$

(c)

Fig. 3. Example of coding: **(a)** the original data matrix; **(b)** an individual of the population; **(c)** the correspondent bicluster

3.2 Fitness Function

The fitness function used to evaluate a given bicluster is given as follows:

$$f(M,N)=\frac{R}{\lambda}+\frac{w_c\cdot\lambda}{|M|}+\frac{w_r\cdot\lambda}{|N|} \qquad (1)$$

where *N* and *M* are the set of rows and columns, respectively, in the bicluster, *R* is called the residue of a bicluster and is calculated as in Eq. 2, λ is a residue threshold (the maximum desired value for residue), w_c is the importance of the number of columns, and w_r the importance of the number of rows. The operator |.| provides the number of elements on a given set. The residue of a bicluster assumes the form:

$$R=\sum_{i,j}\frac{\left(r_{ij}-r_{Ij}-r_{iJ}+r_{IJ}\right)^2}{|N|\cdot|M|} \qquad (2)$$

where r_{ij} is the value of the element (*i,j*) on the original data matrix, r_{Ij} is the mean value of row i, r_{iJ} represents the mean value of column j, and r_{IJ} is the mean value considering all the elements in the bicluster.

With this fitness function we have a ratio between two conflicting objectives: minimizing the residue (variance of elements in the bicluster) and maximizing its volume. Notice that, for a bicluster to be meaningful, it should contain a reasonable number of elements so that some knowledge can be extracted. Also, it is important to maintain some cohesion between its elements.

3.3 The Algorithm

The main algorithm begins by generating a random population of biclusters consisting of just one row and one column, that is, just one element of the data matrix will be used as a "seed" to promote the growth of a local bicluster.

After the initialization, the algorithm enters its main loop. Firstly, the population is cloned and mutated as it will be described later, then if the best clone has a better fitness than its progenitor, it will replace it in the population.

Every **sup_it** iterations the algorithm performs a suppression of very similar biclusters and then inserts new cells. This procedure causes a fluctuation in population size and helps the algorithm to maintain diversity and to work with just the most meaningful biclusters.

The main algorithm of BIC-aiNet is given by the following pseudo-code:

```
Cells = initial_population();
For it=0..max_it do
     For each cell do

             C = clone(cell);
             C = mutate(C);
             If best(C) better than cell
                    cell = C;
             End If
     End For
     If it mod sup_it
             Suppress (cell);
             Insert_new_cells();
     End If
End For
```

3.4 Mutation

The mutation operation consists of simple random insertion/removal procedures applied to the bicluster. Given a bicluster, the algorithm first chooses between insertion or removal operation with equal probability. After that, another choice is made, also with the same probabilities: it will perform the operation on a row or a column. On the insertion case, an element that does not belong to the bicluster is chosen and inserted on the row/column list in ascending order. If the removal case is chosen, it generates a random number that represents the position of the element to be removed from the vectorial representation of the bicluster.

The mutation operator is given as follows:

```
If rand1 < 0.5
       If rand2 < 0.5
               Insert_new_row();
          Else
               Insert_new_column();
       End If
    Else
       If rand2 < 0.5
               Remove_row();
         Else
               Remove_column();
       End If
  End If
```

3.5 Suppression

The suppression step is given by:

```
For each pair i,j of biclusters do
    If $|celli \cap cellj| > \varepsilon*max(size(celli, cellj)/100$
        destroy worst cell
    End If
End For
```

The suppression operation is a straightforward procedure. For each pair of biclusters on the population, it generates the intersection of both sets and counts the number of elements on it, that is, the number of common elements. If this number is greater than a certain threshold ε, it removes the bicluster presenting the worse fitness. After every suppression, the algorithm inserts new cells trying to create biclusters with elements (rows or columns) that still does not belong to any existing bicluster.

4 Experimental Results

This section describes the experiments carried out to evaluate the proposed methodology and to show that interesting information can be extracted from the generated biclusters.

We have applied the methodology to a dataset containing 60 texts and the objective is to group similar documents into one or more categories. Instead of considering the frequency of the attributes, we will represent only presence or absence of a given attribute in the text under analysis. Also, though textual expressions could be considered as a single attribute, we will be restricted to words in their radical form.

Moreover, analyzing the generated biclusters, we will be able to extract the relevant words for each category, extending the usefulness of the methodology. By extracting these keywords, conventional search engines in the internet can be used to find related texts in additional databases.

Considering these two objectives, we show the advantages of biclustering techniques over standard clustering algorithms for text mining.

4.1 Dataset Description and Its Vectorial Representation

The dataset used during the experiments is composed of 60 documents divided into three main categories of 20 texts, and each of these contains two subcategories of 10 texts. Notice that the labels of the texts will not be provided to the biclustering tool, but will be used for further analysis, after concluding the biclustering. The documents were taken from the Brazilian newspaper *Folha de São Paulo* and the categories chosen correspond to sections of the newspaper: Money, Sport and Informatics. Sport news are labeled S and contain two subclasses: Car Racing (S1) and Soccer (S2). Money reports are labeled M and its subcategories are Oil (M1) and International Commerce (M2). The last class, Informatics, is labeled I and is divided into the subcategories Internet (I1) and Technology (I2).

In order to build this dataset, each document had its words extracted and only the radical of each word was taken. After that, a matrix was generated in which each line represents a document and each column presents binary values representing the absence (0) or presence (1) of the related word in the document. Finally, the words that appear in only one document are eliminated from the matrix. After filtered, a total of 1007 words were used as attributes of the 60 documents.

4.2 Parameters Values

This subsection describes the parameters adopted during the experiments as well as their values. Table 1 summarizes these values.

Table 1. Parameters values

Parameter	values
# biclusters	300
# iterations	3000
Residue threshold (R)	1
Row weight (w_r)	5
Column weight (w_c)	19
Suppression Threshold (ε)	80

The algorithm was set to generate up to 300 biclusters during 3000 iterations. As we are dealing with binary data, the residue threshold chosen has a value of "1". The row importance weight and the column importance weight were set empirically in order to achieve a balance between the high volume and the low residue scores. The column importance weight, in this particular case, controls how many words will be used on each bicluster. When the number of columns is high, the results tend to be closer to the ones produced by traditional clustering algorithm.

4.3 Analysis of the Obtained Biclusters

The generated biclusters have residue values in the range between 0.98 and 3.19, meaning that the grouped texts exhibit high coherence among each other. Every document on the dataset belongs to at least one bicluster (though this is not an imposition or necessary condition).

We have observed during the experiments that our algorithm is able to group similar documents efficiently. After the generation of the biclusters, we verified the labels of the texts of a same bicluster and most of them are of the same category. Next, we describe the most significative obtained biclusters and the interesting features that can be extracted from them.

The first bicluster, the one which has the smaller residue value, comprises six out of nine documents belonging to M1 category, indicating the capability of BIC-aiNet to group texts with the same subject. From this bicluster we may extract some dominant words, i.e., words that appear in every document of the bicluster, in order to categorize these texts. The words are: "*Brasil*" (Brazil), "*derivado*" (derived), "*energia*" (energy), "*exportação*" (exporting), "*hipoteses*" (hypothesis), "*Petrobras*", “*petróleo*” (oil) and “*refinaria*” (refinery). With these keywords, popular search engines may be able to categorize on the fly other documents associated with this subject or suggest some “tag words” in order to refine a user search.

The next bicluster has a residue value of 1.19 and contains seven out of nine documents belonging to category M1 and 1 belonging to M2. The same dominant words found on the previous biclusters were found on this one with the addition of the word “*economia*” (economy). The reason for an M2 document being part of this bicluster was that this document was about “steel refinery” and “rationing of energy”, closely related to most documents having “oil” as the main subject.

A bicluster referring to the topic I1 is formed by six out of eleven documents, and it has the following dominant words: “*cadastro*” (filling a form), “*digitar*” (to type), “*golpe*” (scam), “*instalar*” (to install), “*google*”, “*orkut*”, “*internet*”, “*relacionamento*” (relationship), “*mail*”, “*roubo*” (robbery), “*malicioso*” (malicious). It is interesting to notice that an intelligent system could present those words to a user, and the user would point out which subject he is looking for, leading to a more refined search.

Other analyzed bicluster refers to six documents belonging to subclass I1 and one document of M1. The “connection words”, i.e., the words that connected the two subjects were: “*Brasil*”, “programa” (program - can refer to a software or an economic planning), “*público*” (public - as a State service or a free software license), “*serviço*” (service -system service or State service). Here we must point out that the algorithm is sensitive to ambiguous words, which can also become useful as, for example, when a user performs a search on any of these words, an intelligent system may detect that they are ambiguous and ask which of the possible subjects the user is really searching for.

The topic S2 had nine out of its ten texts on a same bicluster connected with the words: “*atleta*” (athlet), “*Brasil*” (Brazil), “*domingo*” (Sunday, the soccer games are usually on Sundays), “*equipe*” (team), “*impulsionamento*” (impulse), “*jogo*” (game), “*segunda*” (second league).

Considering the I2 topic, we could extract significant words as: “*computador*” (computer), “*tecnologia*” (technology), “*software*”, ”*companhia*” (company), “*milhões*” (millions), “*versão*” (version), “*Brasil*” (Brazil), “*US*” and “*Americana*” (American). The four last dominant words are also connected to two documents on M2 category, as most economy related article refers to them.

Words found as dominant of M2 group on its bicluster are: “*bilhões*” (billions), “*milhões*” (millions), “*importação*” (importing), “*mercadoria*” (products), “*recorde*” (record), “*vendas*” (sells), “*US*” and “*Americana*” (American).

So far, it can be seen from these examples that the BIC-aiNet successfully clusters documents strictly belonging to the same subject, documents that have some information in common, and extracts useful information that can be used by intelligent systems in order to refine a search, recommend other readings, finding ambiguities, relating topics on a newsgroup.

In order to visualize the quality of the generated biclusters, a bicluster grouping documents belonging to M1 category is shown on Fig. 4. Each line represents one document and each column a word, the black spots means that the word is present on a given document, the white spots means the absence of a word.

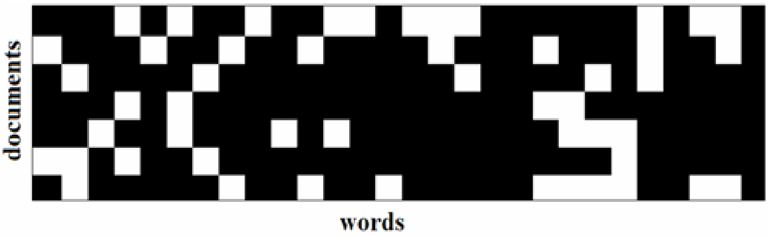


Fig. 4. Example of a bicluster of 7 documents belonging to M1 connected by 28 words

Figure 4 highlights an important aspect of a bicluster algorithm: data compression. In Fig. 5 we can see the original data matrix. As it can be seen, the original matrix is very sparse and of high dimension. Using a biclustering technique, only specific aspects of the whole dataset is taken into account, leading to a more functional clustering outcome.

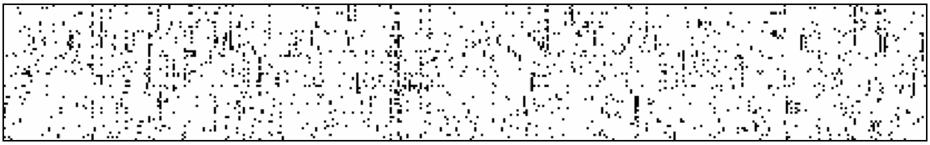


Fig. 5. Original dataset with 1007 words and 60 documents

4.4 Comparative Results

While more rigorous comparisons with other clustering algorithms are still in progress, we present in this subsection the results obtained from a preliminary comparison between BIC-aiNet and the classical K-means clustering algorithm when applied to the same text mining problem. Again, the labels of the texts were not provided to the algorithms. The amount of clusters generated by K-means varied along the experiments. We set K-means to generate 3, 6 and 10 clusters. For all cases, the clusters generated by K-means presented very poor results. By observing the labels of the texts grouped together, we can note that K-means did not separate them efficiently. In all tests, there was one cluster with more than 80% of the texts, while the remaining of the texts was divided into very small clusters.

This outcome suggests that, although the dataset is composed of only 60 documents, it is far from being simple; otherwise, a standard technique such as K-means would have obtained a considerable better performance.

5 Concluding Remarks

This paper introduced a new immune inspired algorithm for biclustering called BIC-aiNet. The proposed biclustering technique is not directly related to the conventional clustering or biclustering paradigm, characterized by the use of the whole dataset, mandatory correlation of all attributes, and the assumption that one item must belong to just one cluster. Instead, we interpret the biclustering problem as multiple bipartition problems. With this flexibility, it is possible to use just some attributes per cluster, making the whole process not only more efficient but also capable of generating information considering several alternative perspectives, possibly useful for classification or decision making in further analysis.

Most texts refer to more than one subject, and "off-topics" happen with some frequency in texts generated by forums or newsgroups. Also some additional information can be extracted from the texts, such as: words that commonly refer to a given subject, words that may refer to more than one subject, how to differentiate ambiguous words, how to classify texts in sub-topics, how to guide a search by using the biclusters to narrow the choices, and so on.

The BIC-aiNet algorithm produces several diverse and high quality solutions simultaneously. Diversity happens when there is little overlap among a set of biclusters, but this overlap is useful when there is several groups that have some features in common. High quality on a bicluster happens when there is little variance on its values (coherent groups).

A dataset of 60 documents extracted from a Brazilian newspaper was generated in order to perform some studies on the information generated by BIC-aiNet. As it was outlined, BIC-aiNet produced coherent clusters and extracted the relevant words belonging to a given subject. BIC-aiNet also correlated different subjects by indicating which words connected them.

As further steps in the research, the experiments will be performed on larger datasets and the information gathered from the experiments will be used to created an intelligent system that will automatically tag a given document and estimate the degree of membership to the subject it was related to, also creating a follow-up list in order to suggest further reading to a user. Also, some post-processing techniques, like variable selection, will be created in order to remove irrelevant words left on the bicluster and eventually include relevant documents left out, making the biclusters even more consistent.

Acknowledgements

The authors would like to thank Universo Online S.A. (UOL – http://www.uol.com.br), CNPq and CAPES for their financial support.

References

1. 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)
2. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proc. of the 8th Int. Conf. on Inteligentt 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: Data Mining: A Heuristic Approach, pp. 231–259 (2001)
4. de França, F.O., Bezerra, G., Von Zuben, F.J.: New Perspectives for the Biclustering Problem. IEEE Congress on Evolutionary Computation, 2768–2775 (2006)

5. 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)
6. Feldman, R., Sanger, J.: The Text Mining Handbook. Cambridge University Press, Cambridge (2006)
7. Goldberg, D., Nichols, D., Brian, M., Terry, D.: Using collaborative filtering to weave an information tapestry. ACM Communications 35(12), 61–70 (1992)
8. Haixun, W., Wei, W., Jiong, Y., Yu., P.S.: Clustering by pattern similarity in large data sets. In: Proc. of the 2002 ACM SIGMOD Int. Conf. on Manag. Data, pp. 394–405 (2002)
9. Hartigan, J.A: Direct clustering of a data matrix. Journal of the American Statistical Association (JASA) 67(337), 123–129 (1972)
10. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Trans. on Computational Biology and Bioinformatics 1, 24–25 (2004)
11. Sheng, Q., Moreau, Y., De Moor, B.: Biclustering micrarray data by Gibbs sampling. Bioinformatics 19(suppl. 2), 196–205 (2003)
12. Symeonidis, P., Nanopoulos, A., Papadopoulos, A., Manolopoulos, Y.: Nearest-Biclusters Collaborative Filtering. In: Proc. of the WebKDD 2006 (2006)
13. 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)
14. Tanay, A., Sharan, R., Shamir, R.: Biclustering Algorithms: A Survey. In: Aluru, S. (ed.) Handbook of Computational Molecular Biology. Chapman & Hall/CRC Computer and Information Science Series (2005)

Defence Against 802.11 DoS Attacks Using Artificial Immune System

M. Zubair Shafiq and Muddassar Farooq

College of Electrical & Mechanical Engineering
National University of Sciences & Technology
Rawalpindi, Pakistan
zubairshafiq@ieee.org, muddassar.farooq@udo.edu

Abstract. In this paper we present an Artificial Immune System (AIS) based security framework, which prevents a number of serious Denial of Service (DoS) attacks. The proposed security framework can counter de-authentication and disassociation attacks. The results of our experiments clearly demonstrate that the proposed framework approximately achieved 100% detection rate with negligible false positive rate. One can conclude from the ROC (Receiver Operating Characteristics) plots of our AIS that its performance approaches 'perfect classification point' at a suitable matching threshold value.

Keywords: 802.11, Network Intrusion Detection, Artificial Immune System.

1 Introduction

Artificial Immune Systems (AISs) are inspired from Biological Immune System (BIS) in vertebrates [1]. BIS protects the body of an organism from foreign antigens. BIS has the remarkable ability to distinguish *non-self* from *self*. AIS maps this feature of BIS to distinguish an anomalous behavior from a normal behavior. AIS learns the normal behavior of a system by tuning an appropriate group of *detectors*. These detectors are used to discriminate the *non-self* antigens from the *self* antigens. Antigens and detectors (antibodies) can be mapped to a n-dimensional real shape-space, where antigen and detector represent two points [1]. The detectors utilize the concept of affinity (affinity between antigen and detector is measured in terms of distance between these points) to differentiate non-self from self. During the initialization phase detectors are generated in a random fashion. However, they are later tuned to *self* using the process of negative selection [3]. The malicious nodes in communication networks can significantly disrupt the normal operations of the networks. AIS based frameworks are ideally suited for NID systems which differentiate the malicious behavior from the normal behavior in the network.

AIS based NIDs have been successfully deployed at the Network layer in wired networks [2,3]. But, to the best of our knowledge, little attention has been paid to the vulnerability analysis of wireless networks. The shared communication

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 95–106, 2007.

medium and the open connectivity policy of wireless networks introduce novel security threats that are not experienced in the wired networks. A number of serious security threats have been discovered at the MAC layer of 802.11b networks.

IEEE 802.11 has become the popular standard for wireless networks in recent years [5]. Most wireless standards deployed today use IEEE 802.11b standard and it is the oldest (launched in July 1999) [5]. With the increasing popularity and usage, several security loopholes and vulnerabilities have been discovered. IEEE 802.11b has been identified for vulnerabilities at Media Access Control (MAC) layer. WEP (Wired Equivalent Privacy) is a classical framework that is deployed at the MAC layer to provide security [5]. In this approach, MAC frame is encrypted using WEP algorithm. Open source tools are available that can break 802.11b WEP. The researchers have also proposed a number of other schemes such as WPA (WiFi Protected Access) and WPA2 (in 802.11i) to cater for security threats in 802.11. These schemes have also failed to provide a satisfactory security level [16,17].

AISs have been used for misbehavior detection in wireless networks [6,8,21]. But almost all these works have focused on routing misbehavior. In this paper we present our comprehensive AIS framework for intrusion detection at the MAC layer. This work is a cardinal step towards the development of a meta-NID based on AIS for misbehavior detection at multiple layers of the protocol stack.

The rest of the paper is organized as follows. In Section 2, we provide a brief introduction to 802.11b wireless networks and discuss different types of vulnerabilities that a malicious node can easily exploit to disrupt the network's operations. In Section 3, we provide a brief review of related work in which AIS has been utilized in NID. We then introduce our AIS based security framework for 802.11b networks in Section 4 and then discuss the experimental results in Section 5. Finally we conclude our work with an outlook to our future research.

2 802.11b Networks

Different wireless standards have emerged to cater for the rapid growth of wireless networks. But IEEE 802.11b is the most popular standard that is deployed in the real world. IEEE 802.11b covers the Media Access Control (MAC) and the physical layer [5].

2.1 Topologies of 802.11b Networks

MAC layer of 802.11b defines two access schemes:

Point Coordination Function (PCF). This scheme is also called infrastructured networks where the complete network is managed by an Access Point (AP). The AP acts as the coordinator in the network. The clients connect to the AP using an authentication and association mechanism.

Distributed Coordination Function (DCF). This scheme is also called adhoc networks which are without coordinator (an AP in case of PCF). In DCF, the clients communicate with each other through the shared channel. The clients, however, have to compete for getting access to the channel. This challenge is not present in PCF in which a station only transmits if it is scheduled a time slot by the AP [5].

2.2 Types of Frames

Three types of frames are exchanged by the communicating nodes at the MAC layer of an 80211b network.

1. Data Frames are used for data transmission.
2. Control Frames are used to control access to the shared medium.
3. Management Frames are only used to exchange management information and hence are not forwarded to the upper layers [5].

2.3 Vulnerabilities in 802.11b Networks

Several vulnerabilities of 802.11b Networks are reported by the authors in [16] which include *Passive Eavesdropping*, *Active Eavesdropping*, *Message Deletion*, *Malicious AP*, *Session Hijacking*, *Man-in-the-Middle* and *De-authentication & Disassociation* attacks.

A malicious node is able to successfully launch the attacks because 802.11b networks provide an attacker the ability to sniff and interpret wireless traffic and spoof the MAC addresses of the AP and the clients. IEEE 802.11b utilizes crypto-based protocols such as WEP and WPA for providing security at the MAC layer. But the authors of [17] have shown that these crypto based solutions are subject to vulnerabilities. As a result, they are unable to provide an adequate security level.

In this paper we have focused on two specific types of DoS attacks which are launched by manipulating *management frames*:

De-authentication Attacks. As already discussed, an attacker can spoof the MAC address of a victim client provided that it has the ability to sniff and interpret the wireless traffic. In de-authentication attacks, an attacker spoofs the MAC address of a victim client and then uses it to send de-authentication frames to the AP (see Figure 1). This attack can significantly degrade the performance of the communication channel between the client and the AP (DoS) because the client, once de-authenticated, must restart authentication/association process again.

A number of schemes have been proposed and implemented for detection and prevention of de-authentication attacks. SNORT uses a '*threshold value of the number of de-authentication frames per unit time*' as a metric for detecting malicious attacker [14]. This scheme applied to bio-inspired techniques, such as genetic programming, results in significantly less detection rates [9,10]. In [18], the authors have used the strategy of delaying the response to de-authentication and disassociation requests. The receiver of the de-authentication frames waits for data frames in the subsequent frames. Valid data frames after

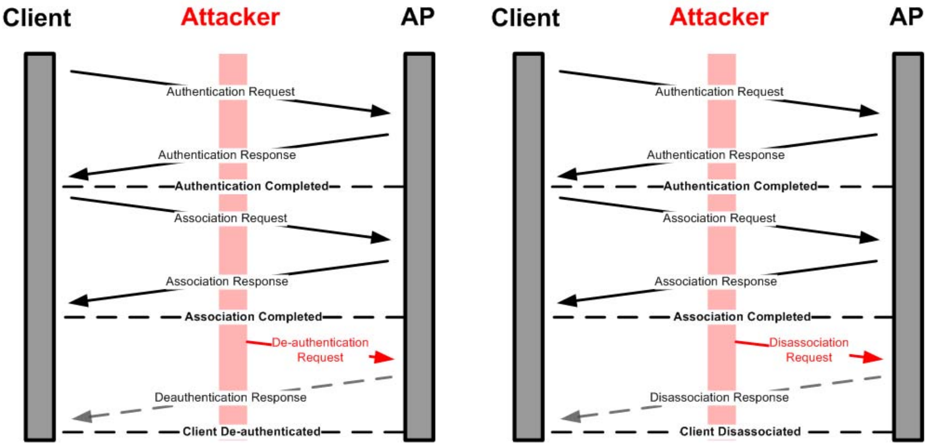


Fig. 1. De-authentication and Disassociation Attacks

the de-authentication frame is an indication of malicious de-authentication attack. This approach, however, results into a number of new vulnerabilities [19]. Another approach utilized the fact in 802.11b networks that the *Sequence Numbers* of the frames vary linearly in case of a normal activity. If an attacker launches the malicious de-authentication attack, then non-linear variations of large magnitudes are observed [6]. The variation in sequence numbers is an important parameter that can act as an indicator to detect de-authentication attack. Fuzzy rules in agent based schemes have also been utilized to detect these random variations in the sequence numbers [7]. In [10,11], the authors have used Genetic Programming to detect de-authentication attacks using such scheme.

Disassociation Attacks. The philosophy behind disassociation attacks is the same as in case of de-authentication attacks. An attacker spoofs the MAC address of a victim node, and then on its behalf sends the disassociation frames to the AP (see Figure 1). Disassociation attacks are less effective in reducing the performance of 802.11b networks because a victim node still remains authenticated with the AP. As a result, it only needs to restart the association process again. Nevertheless, disassociation attacks do cause the disruption in communication (DoS) between victim and AP.

3 AIS for Network Intrusion Detection

AISs have been used for network intrusion detection. The authors in [3] utilized the concepts of AIS such as clonal selection and negative selection for network intrusion detection. Hofmeyr et. al. presented a framework for AIS called *ARTIS* [2]. This framework was specialized for network intrusion detection in Light Weight Intrusion Detection Systems (*LISYS*). LISYS (for wired networks) operated at the Network layer of the OSI Stack [4].

In [6], the authors used a hybrid approach involving AIS for misbehavior detection in wireless ad-hoc networks. Their work is focused on detecting routing misbehavior in DSR (Dynamic Source Routing) due to malware or compromised nodes. In [8], the authors have used AIS exclusively to cater for similar routing misbehavior. In [20], the authors have employed AIS for securing a nature inspired routing protocol, Beehive. Recently they have also proposed an AIS based security framework for a nature inspired wireless ad hoc routing protocol, BeeAdHoc [21]. These works have set the ground for the use of AIS for misbehavior detection in wireless networks.

4 AIS Framework

Our AIS framework consists of two phases: *learning/training* and *detection phase*. During learning/training phase AIS tunes/tolerizes the detectors (antibodies) to the normal behavior of the network utilizing negative selection (*extended thymus action*) [22,23]. This learning/tuning phase takes approximately 30 seconds. During this phase, it is assumed that the system is under normal operating conditions (traffic is training traffic) and any type of anomalous behavior is not experienced during this phase. After learning phase, AIS enters into the detection phase in which it also counters the malicious traffic. During this phase the system detects the malicious traffic and takes countermeasures to neutralize its impact.

Table 1. AIS mapping to 802.11b-MAC

AIS	AIS for 802.11b-MAC
Self-Set	Training traffic set
Non-self Set	Test traffic set
Antigen-1	Antigen consisting of fields from de-authentication frame
Antigen-2	Antigen consisting of fields from disassociation frame
Antibody-1	Detector for de-authentication attacks
Antibody-2	Detector for disassociation attacks
Matching Technique	Euclidean distance matching

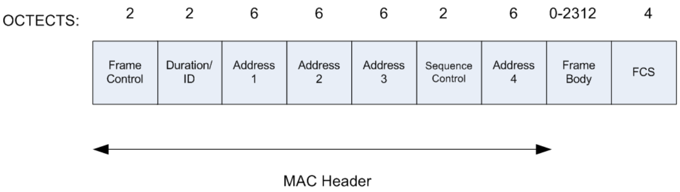
As already mentioned in the previous section, we utilized the same scheme that is based on the observation that the variation in sequences numbers in frames is significantly large in an attack scenario. The AIS mapping used in our AIS model for 802.11b networks is given in Table 1. A MAC frame is mapped on to one of the two types of antigens: type-1 consists of the fields extracted from the de-authentication frame and type-2 consists of the fields taken from the disassociation frame. The Euclidean distance matching technique is used to match antigens to antibodies (detectors). The formula for Euclidean distance is given below [1]:

$$D = \sqrt{\sum_{i=1}^{L} (Ab_i - Ag_i)^2}$$

We have defined two types of antibodies, type-1 and type-2, to cater for two types of attacks. The type-1 detectors are utilized to counter de-authentication attacks whereas type-2 detectors are used to counter disassociation attacks. Their *epitope model* is essentially the same but type-1 and type-2 detectors are tuned using separate sets of training traffic.

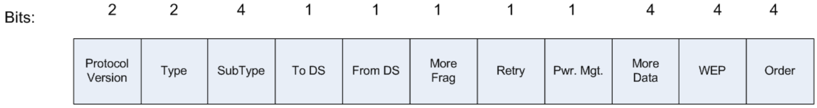
Figures 2 and 3 show the *MAC frame format* and *frame control* field respectively. The sub-fields which are used to vulnerability analysis are:

Type - 00 indicates that it is a *management* frame (2 bits)
Subtype - 1100 indicates that the management frame is a *de-authentication* frame, and 1010 indicates that the management frame is *disassociation* frame (4 bits)
toDS - 0 for management and control frames (1 bit)
fromDS - 0 for management and control frames (1 bit)
Sequence Control - subfield *sequence number* (12 bits)



OCTECTS:	2	2	6	6	6	2	6	0-2312	4
	Frame Control	Duration/ ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	Frame Body	FCS

Fig. 2. MAC frame format [5]



Bits:	2	2	4	1	1	1	1	1	4	4	4
	Protocol Version	Type	SubType	To DS	From DS	More Frag	Retry	Pwr. Mgt.	More Data	WEP	Order

Fig. 3. Frame Control Field [5]

Figure 4 shows the antigen model for our AIS to counter attacks in 802.11b networks. First four fields of the antigen ensure that the appropriate management frame, de-authentication in case of type-1 antigen and disassociation in case of type-2 antigen, is encoded as an antigen. 12 bit *sequence control* field contains the sequence number field of 802.11b MAC frame. The detectors model the average difference in the sequence numbers for every client. It is worth mentioning that difference in the sequence numbers is 1 only in the case when consecutive frames reach the AP without any drops. In case of a lossy environment, the difference in the sequence number even for a client operating under normal conditions will be more than one. There the system needs to learn a threshold value under which the network operation are considered normal. If the distance between the antigen and the detector (antibody) is greater than this threshold value, then a successful match is made. As a result, this match will detect the malicious activity.

BITS:	2	4	1	1	12
	Type	Subtype	toDS	fromDS	Sequence Control

Fig. 4. Antigen Model

5 Results

5.1 Attack Scenario

De-authentication attack (Type-1). Consider the attack scenario, in which 5 clients are authenticated and associated to the Access Point (AP) (see Figure 5). The attacker node spoofs the MAC address of the victim node (node-2) and starts sending the de-authentication frames to the AP with random frequencies. AP will receive the de-authentication frame, and will consider this to be from the victim node. As a result, the connection of the victim node will be invalidated by the AP. In order to communicate again with the AP, node-2 will again have to undergo authentication and association phases (see Figure 1). Hence, the communication between the victim node and AP will be disrupted that might result into poor data transfer rates between the nodes. The level of disruption depends on the frequency of malicious de-authentication frames. In the most severe form of the attack, the victim node will never be able to reach the data transmission phase because as soon as it re-authenticates, next malicious de-authentication frame arrives. The victim node will again be de-authenticated and this malicious cycle repeats itself again. Hence, there is a need to identify and discard the malicious de-authentication frames from the attacker node at the AP. Our AIS will discriminate between the legal and malicious de-authentication frames on the basis of the difference in the sequence numbers of consecutive frames in case of attack.

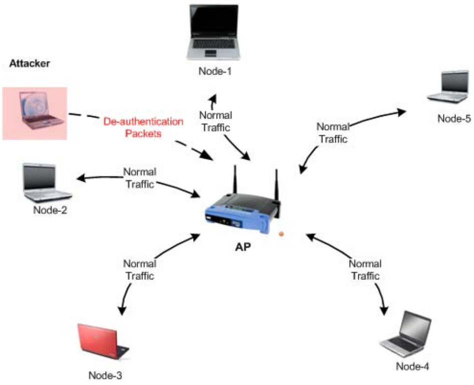


Fig. 5. Attack 1: De-authentication Attack

Disassociation attack (Type-2). Consider the similar attack scenario as shown in Figure 5. The attack node spoofs the MAC address of the victim node (node-2) and starts sending the disassociation frames to AP at random frequencies. AP will disassociate the victim client. In order to communicate victim client would have to undergo association process again, even if it remains authenticated. The reconnection overhead in this case is lesser than the de-authentication attacks as the victim client only has to undergo the association process before being able to enter data sending phase (see Figure 1). Still, disassociation attacks can cause significant disruption and therefore need to be countered. Our AIS will distinguish between legal and malicious disassociation frames on the basis of the difference in sequence numbers of consecutive frames in case of the attack.

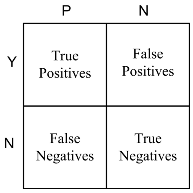
5.2 Dataset Collection

The data-sets (both training and testing) were collected using the tools that are available on the internet such as Ethereal [12] and SMAC 2.0 [13]. Similar tools have also been used in [7] to collect the real traffic. As discussed earlier, the important thing to note in the attack part of testing data-sets is the difference in the sequence numbers when the the attack is launched.

The data-sets which are representatives of the captured traffic were processed by the proposed AIS. AIS requires the training data-set, which is free of attacks, to capture the notion of 'normal'. This training data-set is used to tolerize detectors to normal (*self*) so that they should not detect self-antigens which can result in high number of false alarms. AIS is then exposed to test data-set for detection of malicious de-authentication frames. Using these data-sets, we tested our AIS based NID system. Each run started with a different seed that is used to generate initial detector population randomly. The reported results are an average of the results obtained from 10 independent runs.

5.3 Performance Metrics

Figure 6 shows the 2x2 confusion matrix for the binary classifiers. Four possible outcomes of a classifier are True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). The metrics considered for the evaluation



	P	N
Y	True Positives	False Positives
N	False Negatives	True Negatives

Fig. 6. 2x2 ROC Confusion Matrix (Redrawn from [15])

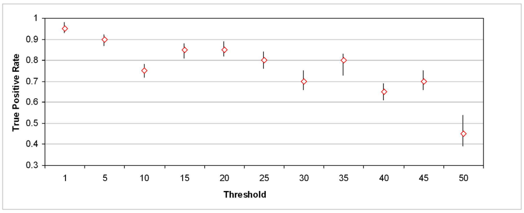


Fig. 7. True Positive Rate for testing data-set (type-1)

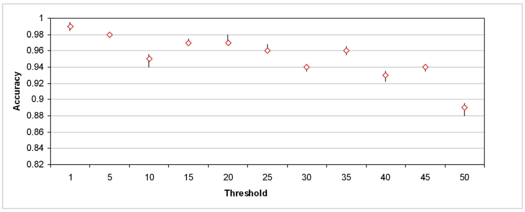


Fig. 8. Accuracy for testing data-set (type-1)

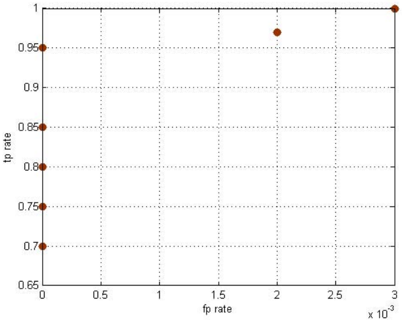


Fig. 9. The performance of our AIS system in the ROC space (type-1)

of AIS are false positive rate *(fp rate)*, true positive rate *(tp rate)* and *accuracy.* These metrics are defined as,

$$tp\,rate = \frac{TP}{P} \qquad fp\,rate = \frac{FP}{N} \qquad accuracy = \frac{TP+TN}{P+N}$$

5.4 Discussion on Results

De-authentication attack. An important parameter of an AIS is a threshold for the match between an antigen and an antibody (detector). We made an interesting observation when we analyzed the effect of varying matching threshold values on the value of false positive rate. We noticed that even for low values of matching thresholds the value of false positive rate was significantly small (as low as 0.001%). Figures 7 and 8 show the plots for *true positive rate* and *accuracy* while varying matching *threshold* value. If we map the values of true positive rate and false positive rate for the type-1 attacks to Receiver Operating Characteristics (ROC) space (see Figure 9) then the performance of our AIS based system approaches the performance of *perfect classification point* [15]. The point at the top-right corner in Figure 9 where system approaches 100% true positive rate is at the threshold value of 4. Therefore, optimal threshold value for matching type-1 antigens and antibodies (detectors) is set to 4.

Disassociation attack. A low value of false positive rate can also be seen for disassociation attacks. This is essentially due to the similar nature of challenges

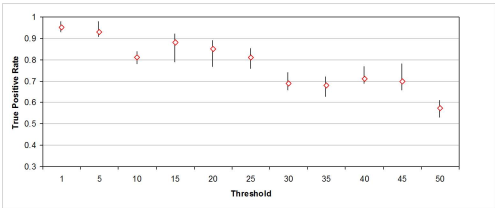


Fig. 10. True Positive Rate for testing data-set (type-2)

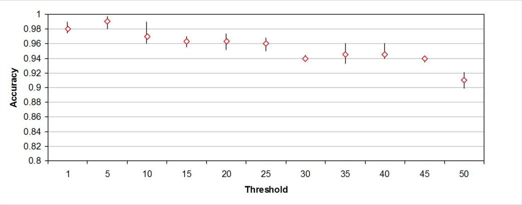


Fig. 11. Accuracy for testing data-set (type-2)

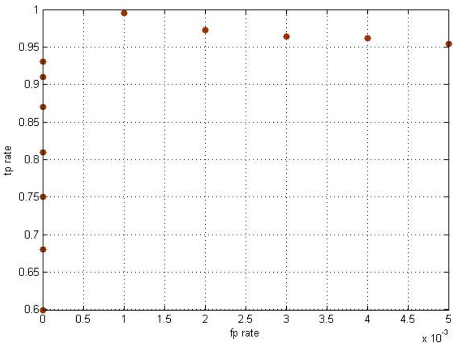


Fig. 12. The performance of our AIS system in the ROC space (type-2)

in both types of attacks. Figure 10 and 11 show the plots for *tp rate* and *accuracy* while varying the value of matching *threshold*. Figure 12 shows the ROC plot for type-2 attack. One can again see that our AIS based system has the performance of a perfect classifier even for type-2 attacks. The optimal matching threshold value for type-2 attacks was found to be 3.

6 Conclusion and Future Work

We have presented a security framework on the basis of principles of AIS for prevention of DoS attacks at the MAC layer. We focused on two specific attacks: de-authentication and disassociation. The results of extensive evaluation of our AIS show that our security framework is able to counter both types of attacks successfully and it has significantly small value of false positive rate. The memory overhead of our AIS based security framework is 3 bytes per detector that is significantly small. Our future objective is to develop a meta-security framework on the basis of AIS for misbehavior detection and prevention in wireless networks at multiple layers.

References

1. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, London (2002)
2. Hofmeyr, S.A., Forrest, S.: Architecture for an Artificial Immune System. Evoloutionary Computation Journal, 443–473 (2000)
3. Kim, J., Bentley, P.J.: Investigating the Roles of Negative Selection in an AIS for NID. IEEE Transactions of Evolutionary Computing, Special Issue on AIS (2001)
4. ISO Standard 7498-1:1994, standards.iso.org/iso/
5. ANSI/IEEE Std 802.11, 1999 edn. (R2003), standards.ieee.org/getieee802/802.11.html
6. Balachandran, S., Dasgupta, D., Wang, L.: A Hybrid Approach for Misbehavior Detection in Wireless Ad-Hoc Networks. Published in Symposium on Information Assurance, New York (June 14-15, 2006)
7. Kaniganti, M.: An Agent-Based Intrusion Detection System for Wireless LANs, Masters Thesis, Advisor: Dr. Dipankar Dasgupta, The University of Memphis (December 2003)
8. Sarafijanovic, S., Le Boudec, J.-Y.: An Artificial Immune System for Misbehavior Detection in Mobile Ad-Hoc Networks with Virtual Thymus, Clustering, Danger Signal and Memory Detectors. In: 3rd International Conference on Artificial Immune Systems, pp. 342–356 (2004)
9. LaRoche, P., Zincir-Heywood, A.N.: 802.11 De-authentication Attack Detection using Genetic Programming. In: Collet, P., Tomassini, M., Ebner, M., Gustafson, S., Ekárt, A. (eds.) EuroGP 2006. LNCS, vol. 3905, Springer, Heidelberg (2006)
10. LaRoche, P., Zincir-Heywood, A.N.: 802.11 Network Intrusion Detection using Genetic Programming. In: GECCO, Workshop Program (2005)
11. LaRoche, P., Zincir-Heywood, A.N.: Genetic Programming Based WiFi Data Link Layer Attack Detection. In: IEEE 4th Annual Communication Networks and Services Research Conference (2006)

12. Ethreal: www.ethereal.com/
13. SMAC: www.klcconsulting.net/smac/
14. Snort- the de facto standard for Intrusion detection/prevention: www.snort.org
15. Fawcett, T.: ROC Graphs Notes and Practical Considerations for Researchers, HP Laboratories (March 16, 2004)
16. He, C., Mitchel, J.C: Security Analysis and Improvements for IEEE 802.11i, Network and Distributed System. In: Security Symposium Conference Proceedings (2005)
17. Arbaugh, W.A., Shankar, N., Wang, J.: Your 802.11 Network has no Clothes. In: Proceedings of the First IEEE International Conference on Wireless LANs and Home Networks, pp. 131–144. IEEE Computer Society Press, Los Alamitos (2001)
18. Bellardo, J., Savage, S.: 802.11 Denial-of-Service attacks: real vulnerabilities and practical solutions. In: Proceedings of the USENIX Security Symposium, pp. 15–28 (2003)
19. Lee, Y.-S., Chien, H.-T., Tsai, W.-N.: Using Random Bit Auhentication to Defend IEEE 802.11 DoS Attacks. ICS, Taiwan (2006)
20. Wedde, H.F., Timm, C., Farooq, M.: Beehiveais: A simple, efficient, scalable and secure routing framework inspired by artificial immune systems. In: PPSN, pp. 623–632 (2006)
21. Mazhar, N., Farooq, M.: BeeAIS: Artificial Immune System Security for Nature Inspired, MANET Routing Protocol, BeeAdHoc. In: ICARIS-2007, Brazil (in Press)
22. Shafiq, M.Z., Kiani, M., Hashmi, B., Farooq, M.: Extended Thymus Action for Reducing False-Positives in AIS based Network Intrusion Detection Systems. In: GECCO-2007, London (in Press)
23. Shafiq, M.Z., Kiani, M., Hashmi, B., Farooq, M.: Extended Thymus Action for Improving the response of AIS based NID against Malicious Traffic. In: CEC-2007, Singapore (in Press)

A Novel Immune Inspired Approach to Fault Detection

Thiago S. Guzella[1], Tomaz A. Mota-Santos[2], and Walmir 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. This paper presents a novel immune inspired algorithm, named *DERA* (Dynamic Effector Regulatory Algorithm), aimed at fault detection and other anomaly detection problems. It integrates immunological research results, pointing out the importance of a special breed of cells (regulatory T cells) in the control of various aspects of the immune system, and includes a mechanism for signalling between cells. Preliminary results of the application of the proposed model to the DAMADICS fault detection dataset are presented, indicating that the proposed approach can attain good results when properly tuned.

Keywords: Fault detection, DAMADICS project, Artificial Immune System, regulatory T cells.

1 Introduction

On an attempt to explore some mechanisms and characteristics of the natural immune system, Artificial Immune Systems [1] (AISs) have been proposed as a novel computing paradigm. An special area of applications of AISs are anomaly detection, or novelty detection, problems (reviewed in [2,3]). These problems are characterized by the availability of data on a single class (or an extreme imbalance between the two classes), usually denominated normal class. Using this data, it is expected that a learning system is capable of determining if a given instance is normal or not. One of the reasons for the special interest of immune inspired systems in this area arises from the direct analogy with the ability of the natural immune system to react against external, harmful agents (nonself or pathogens), while, in most of the cases, remaining unresponsive to internal and harmless components (self). In Immunology, this is referred to as the problem of self/nonself discrimination.

Due to the recent demands for safety monitoring and an ever increasing complexity of the installed industrial plants, fault detection and isolation (FDI) has become a major research area in the last decades [4]. Anomaly detection-based methods can be applied to this area with the special advantage that training requires only data collected during normal operating plant conditions. This is an

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 107–118, 2007.

important feature in most industrial processes, because the introduction or simulation of faults can be extremely costly and time consuming, or even impossible.

The paper is organized in the following way: section 2 briefly discusses some basic immunological concepts, followed by comments on related work in section 3. The novel approach proposed, named *DERA* (Dynamic Effector Regulatory Algorithm) is presented in section 4 and applied to a fault detection benchmark in section 5. Finally, section 6 presents the main conclusions and future directions of this work.

2 The Immune System

The immune system is a complex set of cells and molecules responsible for the maintenance of the host's integrity. An important concept is the self/nonself discrimination, defined as the system's ability to react against external, harmful agents (nonself or pathogens) and spare internal and harmless components (self). This last property, also referred to as self-tolerance, is extremely important, because the immune system has at its disposal destructive mechanisms that must be properly controlled to prevent harm to the host itself.

The immune system can be considered in two parts, the innate and the adaptive systems. The former is composed of cells such as macrophages, dendritic cells and NK (Natural Killer) cells, immediately available to respond to a limited variety of pathogens. The adaptive system, on the other hand, is capable of identifying antigens (molecules that can be recognized by cell receptors and antibodies) never encountered before. This ability is due to the fact that the adaptive system employs a somatically random process to generate cell receptors, allowing the generation of a virtually unlimited number of different receptors. Some of the main cells of this system are B and T lymphocytes. The former are capable of secreting antibodies, whilst the latter are responsible for regulating (regulatory T cells) and stimulating immune responses (helper T cells) and also eliminating host cells infected by intracellular pathogens (cytotoxic T cells). The initial stimulation of B or T lymphocytes happens if the cell receptors bind with sufficient "strength" (affinity) to antigenic features. Unlike B cells, T lymphocytes require that antigenic peptides, coupled to MHC (Major Histocompatibility Complex) molecules, are presented by antigen presenting cells (APCs). Although originated from the same precursors, B lymphocytes develop in the bone marrow, while T cells acquire immunocompetency in the thymus. In addition, cytokines [5] (such as IL-2, IL-10, IFN-γ, among others) are important signalling proteins secreted by various cells of the immune system in response to microbes, antigens and even other cytokines, and play an important role in the control of various aspects of the system. They provide, therefore, means for "communication" between cells, in analogy to the nervous system, where neurotransmitters connect different neurons.

Because the adaptive system randomly generates receptors, mechanisms that would prevent the activation of self-reactive lymphocytes are extremely important. According to the Clonal Selection Theory, proposed by Burnet [6] and that remained as the central idea in Immunology for years, the only mechanism that would assert self-tolerance is a process denominated Negative Selection. During their development, B and T cells would be eliminated if their interaction with self-antigens, presented in the bone marrow and the thymus, respectively, was sufficiently strong (a high affinity interaction). However, it has been known that negative selection is not complete, in the sense that some self-reactive cells eventually escape from deletion, e.g. [7,8], and couldn't, *per se*, ensure self-tolerance [9].

The role of regulatory T cells (formerly known as suppressor T cells) in the prevention of autoimmune diseases has been known for more than three decades (see, e.g., [10]), and has been an intense field of research [11,12]. These cells act as a decentralized self-tolerance mechanism (in contrast to a centralized mechanism, such as negative selection), controlling the activation of other T cells (helper or cytotoxic T cells). So far, experimental evidences point towards regulatory functions based on cytokine secretion and direct cell contact [13].

3 Immune Inspired Approaches to Anomaly Detection

Immune inspired approaches to anomaly detection are usually based on a population of cells (detectors) that analyze an antigen and will be said to recognize it if an affinity measure exceeds a given threshold. For such, data instances are represented as antigens, with a given codification of the attributes. During training, only self antigens (normal data) are presented to the system. One of the pioneering approaches is the Negative Selection Algorithm (NSA), originally proposed by Forrest et al. [14] to monitor changes in binary encoded sequences. By analogy to the process of negative selection that occurs in the thymus and the bone marrow, it generates detectors (analogous to T cells) that will be eliminated if activated by the presented self antigens. After generating the detectors, classification is conducted by presenting an antigen to the generated detectors. If a match between the detector and the antigen occurred, characterized by an affinity greater than the detector's threshold, an anomaly was said to be detected.

Based on this work, the NSA was extended by Dasgupta et al. [15] to deal with real-valued vectors, using the Euclidean distance as affinity measure. Using the real-valued representation, Ji and Dasgupta have proposed the V-Detector algorithm [16], which generates detectors with variable thresholds, so as to maximize the number of nonself antigens that can be recognized. This constitutes the main line of investigation currently, with the proposal of new methods for generating detectors and estimating the coverage of the nonself space (e.g. [17,18,19,20]). Due to fact that some algorithms use normal (self) points to generate nonself detectors, these approaches have been denominated negative detection-based. On the other hand, other approaches will generate self detectors, that have to be capable of matching the self antigens. These positive detection approaches will flag an anomaly if the system is unable to recognize a given antigen. For a detailed discussion of positive and negative detection, see [17,21].

In parallel, there has been work dedicated to incorporating clonal selection in immune inspired anomaly detection systems. Kim and Bentley [22] have proposed the dynamic clonal selection algorithm (DynamicCS) for intrusion detection, where three T cell populations are used: immature detectors, that have to go through a negative selection process, mature detectors, that monitor antigens derived from network traffic, and memory detectors, which have received a confirmation signal by a human expert as being capable of detecting anomalies. Branco et al. [23] have proposed an approach based on selection and mutation to detect faults in induction motors, using modules similar to B and T cells. Ayara et al. [24] developed an adaptive error detection (AED) system for automated teller machines, where immature detectors need feedback supplied by a human expert to become competent. Finally, Dasgupta et al. [25] have developed *MILA* (Multilevel Immune Learning Algorithm), which incorporates innate APCs, B cells, helper and suppressor T cells. The signal supplied by suppressor T cells is extremely important, as all the stimulated B and helper T cells will be suppressed if at least one suppressor T cell has been activated. At the end of the analysis, the activated cells are cloned.

It can be observed that all of these approaches rely on the principle that, in order for a nonself instance to be detected, it is necessary that a detector capable of recognizing it, *per se*, is available. In addition, with the exception of MILA [25], there's no interaction between the detectors in the system. In order to be able to detect all nonself antigens, these systems will rely on covering the shape space [26], a representation of the input space used to distribute the detectors. However, in these cases, the number of detectors tends to be very large, especially for high-dimensional shape spaces. In addition, Stibor et al. [27] have reported that using the Euclidean distance as affinity measure is troublesome in such spaces, making the nonself space coverage problematic. This occurs because the detection region of each detector decreases as the radius is kept constant and the number of dimensions is increased. Based on other arguments, the applicability of the NSA to anomaly detection problems has been questioned [28,29,30]. One of the problems identified is that it is simply a random search procedure, because of the generate-and-test procedure for covering the nonself space [30]. In parallel, some researchers have recently been attracted by Matzinger's Danger Model, and new methods that exploit some of its most important concepts have been developed [31,32]. These methods have been proposed mainly for intrusion detection applications, due to scaling problems previously observed with negative selection algorithms.

4 The Dynamic Effector Regulatory Algorithm (DERA)

The system proposed in this paper, named *DERA* (Dynamic Effector Regulatory Algorithm) is inspired by the work developed in [33], where a simplified model was developed to study a hypothesis for the control of an immune response by regulatory T cells. An interesting point of *DERA* is that it incorporates additional components of the immune system, such as cytokines and regulatory cells. This comes in agreement with a recent work by Stepney et al. [34], which discuss a conceptual framework for bio-inspired algorithms, advocating the use of more detailed biological models. In fact, the consideration of cytokines, an aspect usually overlooked in most AISs, is especially interesting. For example, in the multi-class learning structure inspired by a cytokine network, proposed by Tarakanov et al. [35], interesting results have been obtained in an intrusion detection dataset.

The algorithm proposed uses a population of regulatory and effector cells, combining both "positive" and "negative" detection, and two counters, storing the concentrations of two cytokines in the environment. The underlying principle is that there must be an interaction between the cells in the population before determining if an antigen is self or nonself. In addition, the system is dynamic, possessing a memory, represented by the current cytokine concentrations, so that the classification of an antigen depends on the responses against recently classified instances. Because it doesn't currently include clonal selection, this memory is not antigen-specific. During the training phase, depicted in figure 1, an initial data pre-processing is performed, such as a clustering algorithm, potentially discarding a large number of redundant antigens. This is an important feature, because in anomaly detection problems there's usually a large number of training instances. While this could be used in a positive detection scheme (e.g. [15]), it must be carefully applied, due to the introduction of additional regions that are covered by the self detectors and could be characteristic of anomalies. Using the resultant antigens, the populations are generated and the cytokine counters are initialized to zero. The interactions between cells that take place during classification are described by a set of equations discussed in the following paragraphs.

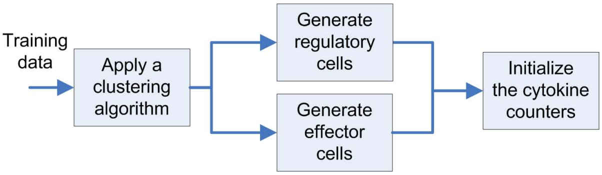

Training data
Apply a clustering algorithm
Generate regulatory cells
Generate effector cells
Initialize the cytokine counters

Fig. 1. Training in the Dynamic Effector Regulatory Algorithm

For the classification of an antigen, illustrated in figure 2, the cytokine concentrations are updated, representing their decay in the environment. The updated cytokine concentrations are given by equation 1, where $\psi(k+1)$ and $\psi(k)$ are the updated and current concentrations, respectively, and ζ is a constant, representing the cytokine decay rate. Obviously, the greater the decay rate, the smaller the influence of the excitations introduced in the system by the previously presented antigens.

$$\psi(k+1) = \psi(k) \cdot (1-\zeta), \quad 0 \leq \zeta \leq 1 \tag{1}$$

When the antigen is presented to the population, the cells receive an stimulus equals to the affinity for the antigen, denoted by α, given by a normalized affinity measure. After the antigen presentation, the n cells with the greatest affinities are selected. If no effector T cell has been selected, no anomaly is detected, and the analysis is finished. Otherwise, the selected cells secrete a stimulatory (in the case of effector cells) or suppressor cytokine (regulatory cells). The selection

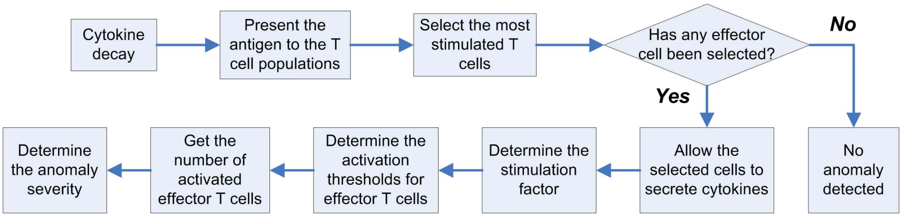

Cytokine decay
Present the antigen to the T cell populations
Select the most stimulated T cells
Has any effector cell been selected?
No
Yes
No anomaly detected
Allow the selected cells to secrete cytokines
Determine the stimulation factor
Determine the activation thresholds for effector T cells
Get the number of activated effector T cells
Determine the anomaly severity

Fig. 2. Classification in the Dynamic Effector Regulatory Algorithm

by the most stimulated cells accounts to an analogy with the natural immune system, where only some of the most stimulated lymphocytes have access to growth factors such as IL-2. Currently, the secretion process is modelled by equation 2, where ψ_c^{sec} is the amount to be secreted by a cell c, α is the affinity and τ is the secretion constant, a positive value that controls the process.

$$\psi_c^{sec}(\alpha) = \tau \cdot \alpha, \ \ \tau > 0 \tag{2}$$

The secreted cytokines are then accumulated in the respective counters. The concentration of the two cytokines determines the system's condition, indicating the predominance of either stimulatory or suppressing effects. This is represented by a coefficient, named stimulation factor (β), that combines the effects of these cytokines. When this factor is negative, suppressor effects are predominant, and only highly stimulated effector cells tend to be activated. When positive, stimulation is dominant, and most effector cells are likely to be activated. This constitutes the framework for *DERA*, where regulatory and effector T cells interact, leading to the ultimate regulating of the activation threshold of the latter cells. The stimulation factor is calculated as a linear combination of the current cytokine concentrations in the system and the amounts secreted in the iteration, according to equation 3, where ψ_s and ψ_r are the concentrations of stimulatory and regulatory cytokines, respectively, and ψ_s^{sec} and ψ_r^{sec} are the concentrations of stimulation and regulatory cytokines secreted in the current iteration. Besides, k_s, k_{ds}, k_r, and k_{dr} are positive constants that have to be adjusted.

$$\beta = k_s \cdot \psi_s + k_{ds} \cdot \psi_s^{sec} - (k_r \cdot \psi_r + k_{dr} \cdot \psi_r^{sec}), \ \ k_s, k_{ds}, k_r, k_{dr} > 0 \tag{3}$$

After determining the stimulation factor, the population of effector cells is analyzed to see which cells will be activated, by comparing the cell-specific adjustable threshold with the affinity for the antigen. For each effector cell, the activation threshold for the current iteration is given by the sigmoid function in equation 4, where Γ_0 is the default, constant activation threshold, β is the stimulation factor and σ is a constant, used to control the steepness of the function. An important feature of equation 4 is that, if the stimulation factor is null, equivalent to an equilibrium between stimulatory and regulatory effects, the activation threshold will be equals to the default threshold. However, if β is positive (negative), the threshold will be decreased (increased). By determining the relative number of effector cells activated by the antigen, which can be considered as a measure of the magnitude of the ingoing artificial immune response, the algorithm outputs the abnormality of a given fault or anomaly. This is an important feature in anomaly detection problems, because the severity of a fault can be used, for example, in an industrial plant, to determine if the process should be interrupted for maintenance. Dasgupta and Gonzalez [36] have proposed a negative selection approach that also outputs the degree of abnormality, but based on the distances from self regions. An special advantage of using equations 2 and 3 in the dynamic model is that it can be shown that the secretion parameters for stimulatory and suppressor cytokines need not be specified, because the stimulation factor depends on the product between the secretion constants and the constants in equation 3. Therefore, it suffices to set the secretion constants as one, and adjust the constants in equation 3.

$$\Gamma(\beta) = \frac{1}{1 + \frac{1-\Gamma_0}{\Gamma_0} \cdot exp(\sigma \cdot \beta)},\ 0 < \Gamma_0 < 1,\ \sigma > 0 \tag{4}$$

It should be noted that, unlike most of the negative detection schemes, *DERA* doesn't rely on the complete coverage of the nonself space. This is due to the fact that the classification of an antigen depends on the interactions between the selected cells, summarized by the stimulation factor, and the affinities of the effector cells. Due to the selection of the most stimulated cells, the detection capabilities of the proposed algorithm depend on the location of the detectors (their receptors), i.e. how close the effector cells are, in the shape space, to the antigen. Finally, the initial selection phase bears some similarity with the k-nearest neighbors (k-NN) algorithm [37]. However, a comparative analysis of *DERA* and k-NN is not included in this paper, and will be investigated later on.

5 Empirical Evaluation

To evaluate the proposed approach, the DAMADICS (Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems) benchmark, proposed by Bartys et al. [38], was selected. It was developed for evaluating and comparing fault detection and isolation (FDI) methods, considering industrial actuators as monitored systems. For testing purposes, the real process data [39] available in the benchmark was selected. It is composed of 33 variables collected during an entire day, with a sampling rate of 1 Hz. The real fault injection has been conducted in three actuators in a sugar factory in Poland, and has been carefully controlled to keep the process operation within acceptable production quality limits. Real data is available for 25 days, ranging from October 29th, 2001 to November 22th, 2001. In this paper, the results obtained when analyzing the data on October 30th, actuator 1, are reported, considering only some of the process variables (namely, P51-06, T51-01, F51-01, LC51-03CV, LC51-03X and LC51-03PV, as in [39]), because these are affected by the occurrence of the fault during this date. The available data was downsampled using a factor of 10, so that every 10th value was used, resulting in 8640 instances per day. The data used for training and analysis was initially pre-processed, using min-max normalization, using the first 4270 instances for training, and the following 4370 instances, for testing. It is important to mention that this procedure was used because it resembles a method for applying anomaly detection methods for fault detection, where the algorithm is initially trained on normal data, e.g. during the beginning of each day, and then used to monitor the plant during the rest of the day. Therefore, the argument of Freitas and Timmis [30] regarding the evaluation of anomaly detection approaches doesn't apply in this case.

For comparison, the V-Detector algorithm [18] was used, using a target coverage of 99%, hypothesis testing with a significance level of 5% and a maximum of 20000 detectors. Results of simply using the data obtained by the application of a clustering algorithm, which is used to generate the population of regulatory cells in *DERA*, are also reported, in a positive detection scheme. The well-known K-Means clustering algorithm [40] was used, executed for a maximum of 1000 iterations. In all the simulations, the Euclidean distance was used as affinity measure. While this can be considered as inappropriate, due to the recent discussions regarding the use of this measure [27,30], in the case of the V-Detector algorithm, the data is low-dimensional. Additionally, and other affinity measures were used, with no clear distinction on the obtained results. Therefore, due to manuscript length restrictions, the authors have chosen to report only the results obtained when using the Euclidean distance.

The effector cell population in *DERA* was randomly generated, with the constraint that, for a new individual to be added, the maximum affinity for the effector cells currently present in the population should be smaller than a given normalized threshold (equals to 0.8). This procedure was used to allow a proper distribution of the effector cells throughout the shape space. Although it is recognized that using such approach is far from ideal, as it constitutes a random search procedure (see [30]), it was used both for simplicity and the unavailability of a more appropriate method, which is still under development. After generating the effector cells, the default threshold ($\mathit{\Gamma}_0$ in equation 4) was set so that each cell would not be activated by the self antigens when $\beta = 0$. If this would result in an invalid threshold ($\mathit{\Gamma}_0 \notin (0, 1)$), the cell would be removed from the population. While this amounts, in a certain way, to covering the shape space, because the thresholds are set to the minimum possible values for each effector cell, it doesn't ensure that the entire space is covered, as required by negative selection algorithms. In the present simulations, 200 effector cells were used, along with 100 regulatory cells (obtained with the K-Means algorithm). The parameters for the dynamic model discussed in section 4 have been selected empirically, with the values $\zeta_s = 0.9$, $k_r = k_s = 2$, $k_{dr} = 3$, $k_{ds} = 4$ and $\sigma = 1$. In the positive detection scheme, obtained as the output of the clustering algorithm, the self threshold of cells in the population was varied, obtaining the classifier's ROC (Receiver Operating Characteristic) curve. In the case of the V-Detector algorithm, the self radius (which determines the threshold of the generated detectors) was also varied. Finally, in the case of *DERA*, the number of selected cells (n) was varied (notice that there's no need to set the activation threshold for regulatory cells).

In addition, the effects of the suppressing cytokine decay rate (ζ_r) have been verified. The ROC curves indicating the performance of each classifier are shown in figure 3, after execution of each test 100 times, including the mean and standard deviation (indicated by the error bars) of the true positive (TP) and false positive (FP) rates. In these curves, the rates for *DERA* are calculated without the application of a threshold in the fault severity to determine if a fault is present. This has been done because the results obtained without thresholding are close to the (TP, FP) pair closest to perfect detection (i.e. $(\mathrm{TP} = 1, \mathrm{FP} = 0)$). In doing so, it is possible to plot each ROC curve as the number of selected cells is varied, which allows one to identify how the detection performance varies as a function of the number of selected cells (equals to 2, 4, 6, 8 and 10 cells, in figure 3). In the presented results, increasing the number of selected cells decreases both detection rates, so that the upper-most points in each curve in figure 3 correspond to selecting the two most stimulated cells per iteration.

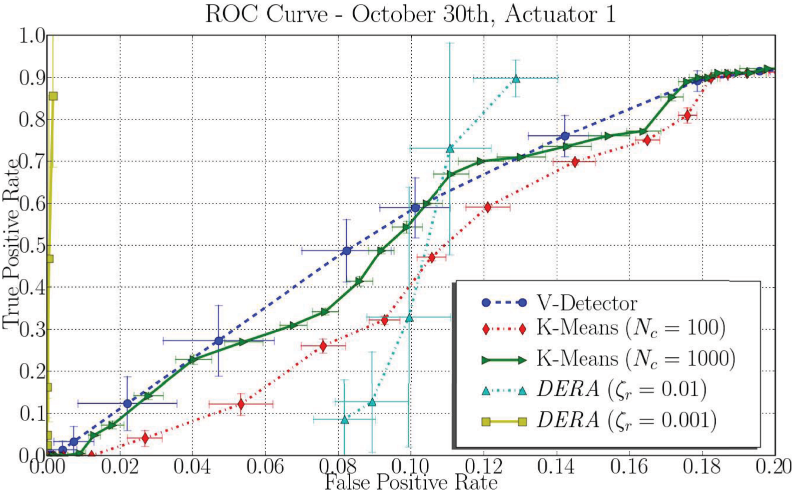


Fig. 3. ROC curves obtained considering the data on October 30th, actuator 1

It can be noticed that the false positive rates tend to be relatively large for most classifiers, reaching 20% for true positive rates of approximately 90%, in this dataset. This is due to a subtle deviation from the normally expected system behavior in the end of the day, which is not characteristic of a fault, according to the dataset. When it comes to the K-Means algorithm, it is clear that increasing the number of clusters leads to better results, at the expense of an increase in the classification time. In terms of number of detectors, the V-Detector algorithm achieved, in the configuration tested, a minimum average number of 516 detectors, with a standard deviation equals to 12, at a normalized self radius equals to 0.001, resulting in the highest true positive and true negative rates. However, as the self radius is increased, the number of detectors needed quickly increases, eventually reaching the maximum number of detectors (20000).In this case, it would not be possible to achieve the desired coverage of the shape space, leading to an increase in the standard deviation in the results. In the case of *DERA*, it can be seen that by decreasing ζ_r, the false positive rate also decreases. The reasons for this behavior are due to a characteristic of the dataset, where the fault is manifested between instances 1611 and 1710 in the testing data. Around instance number 3421, the process starts to deviate from the condition characterized by the system as normal. During this initial phase, some regulatory cells are still among the most stimulated cells, and will secrete suppressing cytokines. Due to the extremely low decay rate of this cytokine, the effect of the secreted cytokines tends to last for a long time, even several periods after the initial deviation, when the process exhibits a large deviation from the initial condition. In this situation, if a fault occurred relatively close to the instants where regulatory cells secrete regulatory cytokines, it could go undetected during an initial period. In other words, the cytokine decay rates, if set to low values, may block the system from quickly responding to a fault. On the other hand, high values may tend to make the system reactive to noise. Therefore, it is expected that there's a value that balances well the ability to quickly respond to faults while remaining unresponsive to noise. Finally, the large standard deviations observed are due to the random generation of effector cells.

6 Conclusions

This paper presented a novel immune-inspired approach to fault detection named *DERA* (Dynamic Effector Regulatory Algorithm), based on a previously modelling work. Unlike negative selection algorithms, that require a complete coverage of the nonself space, the proposed algorithm relaxes this constraint, requiring, instead, an appropriate distribution (but not the coverage) of effector and regulatory cells throughout the space. In addition, it incorporates additional immune system components and dynamics, such as regulatory T cells and cytokines. By combining both regulatory and effector cells, that recognize normal and abnormal operation, respectively, the proposed algorithm possesses a dynamic behavior, mediated by cytokines, and is able to indicate the severity of a fault. An implementation of the proposed classifier was tested on the well-known DAMADICS fault detection benchmark problem as application. The obtained results indicate that, due to a dataset-specific feature, the proposed approach has been capable of attaining considerably lower false positives than other approaches evaluated. This occurred because the regulatory cells would suppress the activation of effector cells. However, if the system is tuned in this way, a slowly growing fault could take a long time to be detected.

The system, as presented in this paper, still has several improvement points. Future work will be conducted to develop an efficient method for generating effector cells, using, for example, evolutionary strategies. In addition, automatic determination of the model parameters is being studied, based on the formulation of an optimization problem. Finally, additional experiments on standard datasets will be conducted to compare the proposed system with other approaches.

Acknowledgments. The authors would like to thank the financial support provided by UOL (www.uol.com.br, process number 20060519110414a), FAPEMIG and CNPq (process number 307178/2004-8).

References

1. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach, 1st edn. Springer, London (2002)
2. Markou, M., Singh, S.: Novelty detection: a review - part 1: statistical approaches. Signal Process 83(12), 2481–2497 (2003)
3. Markou, M., Singh, S.: Novelty detection: a review - part 2: neural network based approaches. Signal Process 83(12), 2499–2521 (2003)
4. Isermann, R.: Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance, 1st edn. Springer, Heidelberg (2005)
5. Abbas, A.K., Lichtman, A.H., Pober, J.S. (eds.): Cellular and Molecular Immunology, 5th edn. W.B. Saunders, Philadelphia, USA (2003)
6. Burnet, F.M.: The clonal selection theory of acquired immunity. Cambridge University Press, Cambridge (1959)
7. Apostolou, I., Sarukhan, A., Klein, L., von Boehmer, H.: Origin of regulatory T cells with known specificity for antigen. Nat. Immunol. 3(8), 756–763 (2002)
8. Schwartz, R.H.: Natural regulatory T cells and self-tolerance. Nat. Immunol. 6(4), 327–330 (2005)
9. Coutinho, A.: The Le Douarin phenomenon: a shift in the paradigm of developmental self-tolerance. Int. J. Dev. Biol. 49, 131–136 (2005)
10. Marx, J.L.: Suppressor T cells: Role in immune regulation. Science 188, 245–247 (1975)
11. Coutinho, A., Hori, S., Carvalho, T., Caramalho, I., Demengeot, J.: Regulatory T cells: the physiology of autoreactivity in dominant tolerance and quality control of immune responses. Immunol. Rev. 182, 89–98 (2001)
12. Sakaguchi, S.: Naturally arising CD4+ regulatory T cells for immunologic self-tolerance and negative control of immune responses. Annu. Rev. Immunol. 22, 531–562 (2004)
13. von Boehmer, H.: Mechanisms of suppression by suppressor T cells. Nat. Immunol. 6(4), 338–344 (2005)
14. Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R.: Self-nonself discrimination in a computer. In: Proc. of the 1994 IEEE Symp. on Res. in Sec. and Priv., pp. 202–212. IEEE Computer Society Press, Los Alamitos (1994)
15. Gonzalez, F., Dasgupta, D.: Anomaly detection using real-valued negative selection. Genetic Prog. and Evol. Mach. 4(4), 383–403 (2003)
16. Ji, Z., Dasgupta, D.: Real-valued negative selection algorithm with variable-sized detectors. Lect. Notes Comput. Sc. 3102, 287–298 (2004)
17. 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)
18. Ji, Z., Dasgupta, D.: Estimating the detector coverage in a negative selection algorithm. In: Proc. of the GECCO, pp. 281–288. ACM Press, New York (2005)
19. Ji, Z.: A boundary-aware negative selection algorithm. In: Proc. of the 9th IASTED ASC. Acta Press (2005)
20. Shapiro, J.M., Lamont, G.B., Peterson, G.L.: An evolutionary algorithm to generate hyper-ellipsoid detectors for negative selection. In: Proc. of the GECCO, pp. 337–344. ACM Press, New York (2005)

21. Garrett, S.M.: How do we evaluate artificial immune systems? Evol. Comput. 13(2), 145–178 (2005)
22. Kim, J., Bentley, P.J.: Towards an artificial immune system for network intrusion detection: An investigation of dynamic clonal selection. In: Proc. of the IEEE CEC, pp. 1015–1020. IEEE Computer Society Press, Los Alamitos (2002)
23. Branco, P.J.C., Dente, J.A., Mendes, R.V.: Using immunology principles for fault detection. IEEE T Ind. Electron 50(2), 362–373 (2003)
24. Ayara, M., Timmis, J., de Lemos, R., Forrest, S.: Immunizing automated teller machines. Lect. Notes. Comput. Sc. 3627, 404–417 (2005)
25. Dasgupta, D., Yu, S., Majumdar, N.S.: MILA - multilevel immune learning algorithm and its application to anomaly detection. Soft Comput. 9, 172–184 (2005)
26. Perelson, A.S.: Immune network theory. Immunol Rev. 110(1), 5–36 (1989)
27. Stibor, T., Timmis, J., Eckert, C.: On the use of hyperspheres in artificial immune systems as antibody recognition regions. Lect. Notes Comput. Sc. 4163, 215–228 (2006)
28. Stibor, T., Mohr, P., Timmis, J., Eckert, C.: Is negative selection appropriate for anomaly detection? In: Proc. of the GECCO, pp. 321–328. ACM Press, New York (2005)
29. Stibor, T., Timmis, J., Eckert, C.: A comparative study of real-valued negative selection to statistical anomaly detection techniques. Lect. Notes Comput. Sc. 3627, 262–275 (2005)
30. Freitas, A.A., Timmis, J.: Revisiting the foundations of artificial immune systems for data mining. IEEE Trans. Evol. Comput. (in press) DOI: 10.1109/TEVC.2006.884042 XX (2006) XX
31. Sarafijanović, S., Le Boudec, J.Y.: An artificial immune system approach with secondary response for misbehavior detection in mobile ad hoc networks. IEEE Trans. Neural Networks 16(5), 1076–1087 (2005)
32. Greensmith, J., Aickelin, U., Cayzer, S.: Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection. Lect. Notes Comput. Sc. 3627, 153–167 (2005)
33. Guzella, T.S., Mota-Santos, T.A., Uchôa, J.Q., Caminhas, W.M.: Modelling the control of an immune response through cytokine signalling. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 9–22. Springer, Heidelberg (2006)
34. 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)
35. Tarakanov, A.O., Goncharova, L.B., Tarakanov, O.A.: A cytokine formal immune network. In: Capcarrère, M.S., Freitas, A.A., Bentley, P.J., Johnson, C.G., Timmis, J. (eds.) ECAL 2005. LNCS (LNAI), vol. 3630, pp. 510–519. Springer, Heidelberg (2005)
36. Dasgupta, D., Gonzalez, F.: An immunity-based technique to characterize intrusions in computer networks. IEEE Trans. Evol. Comput. 6(3), 281–291 (2002)
37. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press, London (1990)
38. Bartyś, M., Patton, R., Syfert, M., de las Heras, S., Quevedo, J.: Introduction to the DAMADICS actuator FDI benchmark study. Control Eng. Pract. 14(6), 577–596 (2006)
39. Bartyś, M., Syfert, M.: Data file description. In: manuscript on (2002), http://diag.mchtr.pw.edu.pl/damadics/
40. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: LeCam, L., Neyman, J. (eds.) Proc. of the 5th Berkeley Symp. on Math. Stat. and Prob. 1, 281–297 (1967)

Towards a Novel Immune Inspired Approach to Temporal Anomaly Detection

T.S. Guzella[1], 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 conceptual paper, we report on studies and initial definitions of an immune-inspired approach to temporal anomaly detection problems, where there is a strict temporal ordering on the data, such as intrusion detection and fault detection. The inspiration for the development of this approach comes from the sophisticated mechanisms involved in T-cell based recognition, such as tuning of activation thresholds, receptor down-regulation, among others. Despite relying on low affinity and highly degenerate interactions, the recognition of foreign patterns by T cells is both highly sensitive and specific. Through a proper understanding of some of these mechanisms, this work aims at developing an efficient computational model using some of these concepts.

Keywords: Artificial Immune Systems, Tunable activation threshold hypothesis, T cells, Temporal anomaly detection.

1 Introduction

The immune system is a complex set of cells, antibodies and signalling molecules. It employs several mechanisms to maintain the integrity of the host with a great efficiency. Some of its interesting features, from a computational point of view, is its distributed characteristic, along with learning and self-organizing abilities. Based on such defining features, it has inspired the development of Artificial Immune Systems (AISs), a novel soft computing paradigm [1], by exploring certain mechanisms of this system.

In this paper, we present initial steps towards a novel approach to temporal anomaly detection. The difference in regard to the more general anomaly detection problems (surveyed in [2,3]) is that there is a temporal relationship between data instances collected during a given time interval. In other words, there's a strict temporal ordering in the data being analyzed. Some examples of temporal anomaly detection problems are intrusion detection (recently reviewed in depth in [4]) and fault detection in dynamic systems [5], problems that have been of particular interest to the application of AISs. For the development of this model,

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 119–130, 2007.

there has been motivations to understand in greater details the aspects involved in the recognition of foreign peptides by T cells, such as the processing of antigens for presentation in MHC molecules (as peptide/MHC, or pMHC complexes), the intricate signalling machinery of T cells, and the tunable activation threshold hypothesis, proposed by Grossman and colleagues [6,7,8,9], according to which T cells would adjust their responsiveness during the developmental stage in the thymus and in the periphery.

This paper is organized in the following way: section 2 describes some of the mechanisms involved in T-cell recognition of antigens, also briefly surveying some modelling studies in this area, while section 3 comments on related work on the area of AISs. In sequence, section 4 presents the initial steps towards a coherent T-cell inspired anomaly detection approach, followed by the conclusions and future directions in section 5.

2 Inspirations from Immunology

2.1 $\alpha\beta$ T Cells

In the adaptive immune system, T cells play a major role. These cells are divided in two subsets: $\alpha\beta$ and $\gamma\delta$ T cells, where the latter are still not very well understood, despite recent advancements. For this reason, this paper focuses on $\alpha\beta$ T cells, which will be referred to as simply "T cells".

Helper T cells activate stimulated B cells, while cytotoxic T cells eliminate host cells infected by intracellular pathogens, and regulatory T cells regulate the activation of other B and T cells. Each T cell has, on average, 30000 receptors (TCRs) [10], expressed on the membrane, which are capable of recognizing antigen-derived peptides coupled to class I (expressed by almost all cells in the body) or class II MHC (Major Histocompatibility Complex) molecules, the latter being presented by antigen presenting cells (APCs). In general, MHC molecules are not able to discriminate between self-derived and foreign peptides [11], resulting in a small number of foreign peptides being presented in conjunction with a large number of abundant self peptides. These cells are produced in the bone marrow, and acquire immunocompetency in the thymus, after being submitted to two selection processes. The first process, denominated positive selection, assures that the developing thymocytes are MHC restricted (capable of recognizing peptides coupled to a single MHC isoform); negative selection, on the other hand, eliminates T cells that sustain a high affinity interaction with self ligands [12] presented in the thymus.

Binding between a TCR and a peptide/MHC (pMHC) leads to the transduction of a signal, and subsequent triggering of the receptor. An interesting aspect is that triggered receptors are internalized (TCR down-regulation) which is thought to allow efficient interaction with ligands expressed in low concentrations [11]. The immunological synapse, a structure that mediates the interaction between a T cell and an APC, has been reported to allow T cells to communicate with several APCs at the same time, and select the one providing the strongest stimulus [13]. Another important aspect that has emerged is the need for sustained signalling [14], and maintaining serial encounters with several APCs [15], related to the stochastic nature of TCR triggering (see, e.g. [16]).

Based on the ability to detect small amounts of foreign peptides and do not respond to a large number of self peptides, Davis et al. [17] classify T cells as sensory cells. In fact, this sensory characteristic has interesting similarities with the nervous system, such as the coupling between sensitivity and environmental conditions [18]. George et al. [16] review how specificity (recognition of small variations in the pMHC complex presented) and sensitivity (detecting low concentrations of foreign pMHCs) are achieved in T cell recognition, despite the generally low affinity interaction and its degeneracy. At the TCR level, the serial engagement model [19] (in which a single pMHC is capable of triggering several receptors) explains the sensitivity, while the kinetic proofreading model [20] (the need for an interaction between the TCR and the pMHC to last longer than a receptor triggering threshold in order for the TCR to be triggered [21,16]) explains the specificity.

Due to the major role played by T cells in adaptive immune responses, it has motivated the development of several theoretical studies, some of which are summarized in [22,23,24]. Noest [25] analyzed the requirements for optimal detection performance, based on the signal detection theory, assuming that APCs allow T cells to sample a variety of ligands. In that study, it was observed that T cells possess various characteristics that would allow optimal detection, such as clonal diversity, receptor specificity, serial triggering and tuning of activation thresholds, among others. Using the concept of activation curves, van den Berg et al. [26] have studied how the T cell repertoire is able to respond to low concentrations of foreign peptides coupled to MHC molecules, while minimizing the responsiveness to self peptides, based on minimizing (maximizing) the probability of initiating a response in the absence (presence) of foreign peptides. In that model, the TCR triggering rate was considered as a measure of avidity (a combination of ligand concentration and binding affinity). They have concluded that a repertoire satisfying these conditions is possible, even though being based on low affinity and highly degenerate recognition. In that model, negative selection would allow the recognition of foreign peptides in lower concentrations. Following this, they have analyzed the effects of presenting peptides that are variants of a foreign peptide [21], which can act as agonists (capable of evoking the full range of cellular responses) or antagonists (that inhibit the effects of other ligands). An interesting conclusion was that, by varying the number of receptors expressed, a T cell could be tuned to different ligands, being capable of distinguishing between the ligand's triggering rate and its presentation level. Later on, van den Berg and Rand [27] studied the role of MHC presentation. It was concluded that immune efficacy (represented by the separation between triggering rate distributions in the absence and in the presence of foreign ligands) is maximized when each T cell clonotype is capable of interacting with a single MHC isoform, presenting a low diversity of foreign peptides.

2.2 The Tunable Activation Threshold Hypothesis

The tunable activation threshold (TAT) hypothesis was proposed by Grossman and colleagues [6,7,8,9], discussing the possibility of T cells (and also B cells) dynamically tuning their thresholds in response to recently experienced stimuli. It, therefore, proposes the idea of a T cell as an adaptive signal integrator, where the inputs are the signals registered by the TCRs, and the output can be characterized as a cellular state. Tuning would allow the cell to measure relative, instead of absolute, changes in the input signal magnitude, and would act so that a cell under persistent stimulus should not be activated [28]. As commented by Grossman and Paul [8], a controlled degree of autoreactivity is essential in the immune system. In this sense, self recognition could be a way for the immune system to "tune" its sensitivity (e.g. [11,29]). In fact, a similar feature has been described in the nervous system: in the mammalian neocortex, Stetter [30] analyzes the presence of a continuous background neuronal activity that would allow the dynamic tuning of neocortical circuits. An important proposal of the hypothesis is that cellular anergy (a particular state where the cell remains unresponsive to stimuli) would be characterized by an elevated activation threshold, and may have important roles in the control of autoimmunity and immune responses against foreign antigens. In addition, tuning would entail each T cell with a short-term memory, which has been suggested to facilitate long-term memory at both cell and population level [6].

In the hypothesis, two modes of response are considered [6]. Supra-threshold responses are characterized by a rapid increase in the antigen concentration, leading to a complete elimination of the antigen. On the other hand, sub-threshold responses would be invoked in the case of a chronic or residual infection, and could lead to several cellular outcomes, such as tuning the activation threshold, promoting the viability and excitability of the cell, among others. Later on, possible roles for negative and positive selection within this hypothesis have been discussed [7]. The former would lead to the elimination of unnecessary cells, which would not be activated in the periphery, due to the extremely high activation thresholds. Therefore, negative selection would not be responsible for the elimination of auto-reactive cells. Positive selection, on the other hand, is related to a viability maintenance threshold, indicating that a cell under development must be capable of receiving signals from the environment.

Due to its inherent dynamical nature, the TAT hypothesis has inspired several modeling investigations. Sousa [28] studied the implications of threshold tuning on the homeostasis of T cells. It was concluded that tuning is unable, *per se*, to allow the persistence of autoreactive cells, requiring additional mechanisms to accomplish this, but is able to prevent such cells from expanding and inflicting autoimmunity in the periphery if the thymic output rate of such cells is relatively small. An additional requirement would be that these cells were pre-adapted to self antigens presented in the thymus. A more recent study of this model has been presented by Carneiro et al. [31], which have introduced simplifications to obtain analytic expressions. It was used to study the effects of an adoptive transfer of peripheral T cells, and concluded that it tolerance would have to be induced by some interaction between populations not considered in that model.

In another line of investigation, van den Berg and Rand [32] have analyzed the biochemical plausibility of the hypothesis, proposing two mechanisms through which threshold tuning could take place. They have suggested that threshold tuning may only take place in the absence of danger signals [33], because if a T cell, once activated by a ligand, were allowed to adapt to the received stimulation, it would become less sensitive to that ligand. In addition, an important conclusion was that threshold tuning would allow the immune system to maximize the sensitivity to nonself antigens uniformly across the T cell repertoire and the lymphoid tissues.

Scherer et al. [34] compared a model where the threshold is set for each cell individually, instead of using a constant value for all cells, assumed to have evolved to an optimal value. In that model, the activation threshold for each cell was given by the maximum affinity for self ligands presented during negative selection, so that the resultant cells would not be activated by such ligands. It is important that such model is a special case (with $d\alpha(t)/dt \geq 0$, $\forall t$, where $\alpha(t)$ is the threshold at time instant t) of the TAT hypothesis, as Grossman and colleagues proposed that the threshold is tuned depending on the recent stimulation history of the cell. It was verified that tuning is more flexible, allowing the generation of repertoire that are more reactive to nonself peptides than those generated by constant threshold mechanisms, provided that the repertoire size is large.

3 Related Work

This section comments on related immune-inspired approaches, that incorporate some of the biological mechanisms previously discussed. Readers interested in more details should consult recent reviews, such as [35,36,37].

Despite being a fundamental aspect of T cell recognition in the biological immune system, the analogy of antigen processing and presentation coupled to MHC molecules hasn't been employed in most artificial immune systems, in which T cells bind directly to antigens. Hofmeyr and Forrest [38] have introduced the concept of a permutation mask, to decrease the number of undetectable patterns in negative selection algorithms, inspired by MHC-coupled presentation of peptides. Dasgupta et al. [5] have proposed MILA (Multilevel Immune Learning Algorithm), where T cells are defined as low-level recognition entities, attempting to model MHC processing. However, the pseudocode available in [5], aimed at anomaly detection problems, does not consider this aspect, allowing B and T cells to bind in the same way, where the affinity is based on the Euclidean distance. In addition, more sophisticated aspects of T-cell based recognition, such as the expression of multiple receptors and receptor down-regulation, are yet to be incorporated into an immune inspired algorithm.

In regard of the TAT hypothesis, there does not seem to be any algorithm designed to directly explore it. Most immune-inspired algorithms employ the concept of a threshold, which is compared with some state of each cell or the system, to determine if a given procedure, such as cell activation, proliferation or updating the system will take place. Among such approaches, some are capable of adapting such thresholds based on the features of the target application, instead of including them as user set parameters.

An algorithm aimed at extracting fuzzy classification rules, named IFRAIS (Induction of Fuzzy Rules with an AIS) has been proposed by Alves et al. [39]. In that case, the affinity is given by the degree of matching between a rule (antibody) and an example (antigen), and the latter satisfies the relation implied by the former if this degree is greater than an affinity threshold. The system employs an adaptive mechanism that adjusts each affinity threshold, so as to maximize the fitness of each rule. The Artificial Immune Recognition System (AIRS) [40] initially determines a threshold based on the data, given by the average affinity for the antigens (the training data). This threshold is then kept constant during the training phase. In some immune network based approaches, e.g. [41,42] (reviewed in [43]), a threshold for cloning some network nodes is determined, in each iteration, based on the responsiveness of the nodes.

Finally, some negative selection algorithms have incorporated, intuitively, thresholds adjusted during the detector generation phase, depending on the detector's affinity for the self antigens. Gonzalez and Dasgupta [44] have proposed an anomaly detection algorithm based on the generation of detectors in the non-self space. The threshold for each detector is generated based on the median of the distances to the k-nearest self points. In the V-Detector algorithm [45], another negative selection based approach, the distance, analogous to the threshold, is determined based on the distance between a candidate detector and the closest normal (self) point. Therefore, the activation threshold is determined for each detector, depending on the affinities with the normal data, so that the detector is unable to be activated by these points. In fact, this approach is very similar to the model studied by Scherer et al. [34], with an additional constraint on the maximum value of the threshold (related to the self radius in [45]).

This brief listing of variable threshold algorithms is not extensive, but primarily aimed at indicating that some approaches incorporate some form of adjustable threshold, on an intuitive basis. In the network based approaches, threshold tuning asserts the continuous learning capabilities of the network. Finally, one should notice that, out of the aforementioned algorithms, with the exception of [41,42,39], the threshold is determined during the training phase and remains constant thereafter.

4 Initial Architecture for a T-Cell Inspired Anomaly Detection System

In this section, the first steps towards the architecture for an immune-inspired model including some of the features discussed in section 2 are presented. The development of this model is motivated by three aspects. First, the wide use of several computational and engineering terms in describing some aspects of T-cell based recognition (e.g. decision making, noisy environments [11]). In addition, the remarkable ability of these cells to achieve high specificity and sensitivity, even though based on low affinity and highly degenerate interactions, to detect low amounts of foreign peptides in a vast sea of self ligands (analogous to noise). Finally, because, there does not seem to exist, to the knowledge of the authors, any work in the area of AISs dedicated to using more detailed models of T cells. It is important to notice that, in contrast, in the area of neural networks, there are several researchers investigating more realistic neuron models [46,47].

Recently, Stepney et al. [48] have presented a conceptual framework for bio-inspired algorithms, suggesting the use of more detailed biological models to develop algorithms. Because AISs are still a relatively new area of research, there is a lot of unexplored potential, especially regarding the incorporation of additional theories and concepts from Immunology. In recent years, several efforts in this line have emerged [49,50,51,52]. One promising example is the Dendritic Cell Algorithm, proposed by Greensmith et al. [51], inspired by the Danger Model [33]. The model presented in this paper can be considered as yet another attempt to incorporate additional features of the natural immune systems in AISs. However, due to the fact that this work is still in an initial phase, this section reports on some of the aspects incorporated so far. The architecture presented here is initially aimed at temporal anomaly detection problems, where there is an strict temporal relationship between data. The starting point is that the operating conditions are sampled from a given process (e.g. a computer network, an industrial plant) as an antigen $\boldsymbol{x}_k$, as illustrated in figure 1.

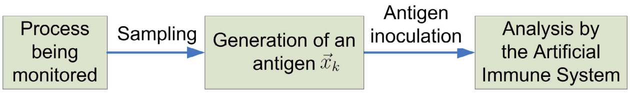

Process being monitored
Sampling
Generation of an antigen $\vec{x}_k$
Antigen inoculation
Analysis by the Artificial Immune System

Fig. 1. Representation of sampling data (antigens) from the process being monitored

The model adopted for each T cell attempts to consider a more biologically plausible representation, incorporating multiple receptors and, therefore, extending the notion of a cell as a multi-input processing unit combining various signals received from the environment. In addition, as will be presented in sequence, this is related to the temporality of the target problem. Thus, T cells can be considered as depicted in figure 2. The TCR triggering rate experienced by the i-th T cell at time instant k, denoted as $W_i(k)$, is given by equation 1, based on the generalization of a model developed by van den Berg et al. [26], where T_{ij} is the mean dwell-time of the interaction between the cell's TCR and the j-th pMHC, T_R is a receptor threshold, $n_T(i,k)$ is the number of TCRs expressed by the i-th cell at time instant k (therefore, taking TCR down-regulation into consideration, according to some yet unspecified kinetics) and f_i is a binding function, to be chosen so that the signal enlicited by a pMHC is maximum when $T_{ij} = T_R$.

$$W_i(k) = \sum_{j=1}^{n_T(i,k)} f_i(T_{ij}, T_R) \tag{1}$$

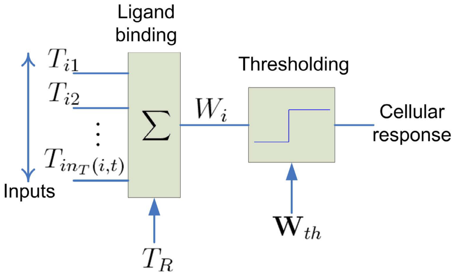


Fig. 2. T Cell Model

In this model, allowing multiple thresholds ($\boldsymbol{W}_{th} \in \Re^{n_S}$, where n_S is the number of states considered) reflects the fact that there are different thresholds for some cellular responses, ranging from the induction of self-renewal division through cytokine secretion, as extensively discussed by Grossman and colleagues [6,7,8,9]. In considering that T cells feature a highly adjustable and flexible signalling machinery, T_R, T_{ij} and $\boldsymbol{W}_{th}$ can be adapted, depending on some conditions.

An important feature of T-cell recognition is MHC restriction, which can be considered as a feature selection or extraction process. In the proposed model, MHC restriction is modelled by considering a given number of separate T cell sub-populations, where each sub-population is allowed to interact with only a single feature (a peptide) of an antigen, ignoring all the remaining peptides. Therefore, in this model, the presentation of antigen-derived peptides coupled to MHC molecules is represented by the application of feature selection/extraction algorithms [53]. Although it does not consider the competition between peptides for loading in MHC molecules, as in the biological immune system, it is believed that it captures the essence of the process: having segregated T cell populations, that can recognize only a certain feature of an antigen. Also, it entails an additional degree of parallelism, at the level of sub-populations, in addition to that at the detector level.

The basic architecture for the model is presented in figure 3. By introducing the abstraction of separate T cell sub-populations that process a feature derived from an antigen, the model does not incorporate positive selection, as it is assumed that each sub-population is generated separately, and there is no need to enforce MHC restriction. Feature selection/extraction is applied to an antigen $\boldsymbol{x}_i$, and generates n_i peptides $\boldsymbol{X}_i^1, \boldsymbol{X}_i^2, \ldots, \boldsymbol{X}_i^{n_i}$, where n_i is the number of MHC isoforms considered. In introducing a step where the cells in the population interact among themselves, the model incorporates the inherent connectivity among cells in the immune system. For example, in the tunable activation threshold hypothesis, it has been suggested that some cells may be capable of raising the thresholds of other cells in the neighborhood [6].

The temporality in the anomaly detection problem is captured, on one side, from the retention of the w most recently injected antigens in the system, through the application of a non-overlapping window. Therefore, the patterns analyzed

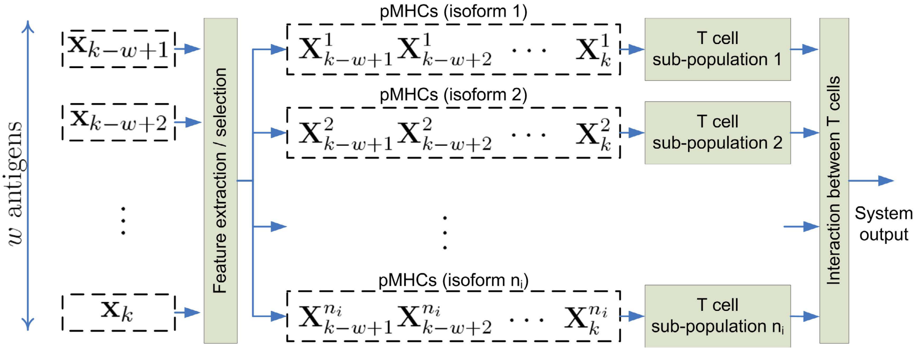

w antigens
$\mathbf{x}_{k-w+1}$
$\mathbf{x}_{k-w+2}$
$\mathbf{x}_{k}$
Feature extraction / selection
pMHCs (isoform 1)
$\mathbf{X}^1_{k-w+1}\mathbf{X}^1_{k-w+2} \cdots \mathbf{X}^1_{k}$
pMHCs (isoform 2)
$\mathbf{X}^2_{k-w+1}\mathbf{X}^2_{k-w+2} \cdots \mathbf{X}^2_{k}$
pMHCs (isoform n$_i$)
$\mathbf{X}^{n_i}_{k-w+1}\mathbf{X}^{n_i}_{k-w+2} \cdots \mathbf{X}^{n_i}_{k}$
T cell sub-population 1
T cell sub-population 2
T cell sub-population n$_i$
Interaction between T cells
System output

Fig. 3. Architecture of the proposed approach

by the system at a time instant k are a total number of $n_i w$ peptides, derived from antigens $\boldsymbol{x}_{k-w+1}, \boldsymbol{x}_{k-w+2}, \ldots, \boldsymbol{x}_k$. The advantage of focusing on temporal problems arises from the possibility of the system reacting to features which, even though being derived from different antigens, show a time correlation that could be interesting. As an overly simplified example, if peptides, representing the temperature of a system, are derived from consecutive antigens, and indicate a subtle increase in the monitored temperature, then the system could find this information indicative of a fault. The use of this window has some similarities with the approach used by Gonzalez and Dasgupta [44] and Dasgupta et al. [5] to detecting anomalies in time series, where, differently from the model discussed here, each antigen is generated by the application of the window. The difference in the presented model is that, due to the fact that T cells have multiple receptors, they are capable of receiving stimuli from different pMHC at each time step, depending on criterion used to determine which of the pMHCs will bind to which T cells. The characteristic of receiving multiple signals correlates better, in the view of the authors, to how natural T cells receive environmental signals, instead of the single receptor model considered in most AISs. In addition, an additional parameter that needs to be specified is the concentration of each ligand, i.e. how many copies of each peptide are introduced in the system by feature selection/extraction.

Finally, one last aspect must be commented on. Considering the mechanisms considered so far (namely, MHC presentation, multiple receptors, receptor down-regulation and tunable thresholds), and, based on the architecture depicted in figure 3, the novelty of the model discussed (when compared, e.g., with a neural network with tunable thresholds) arises from two main features. The first is not present in neurons: by varying the number of receptors, each T cell would be capable of receiving signals from a different number of sources at each time instant. Another feature is the fact that each cell has several possible responses (such as proliferation, activation, or cytokine secretion), each one characterized by a threshold, where each response could affect other cells in different ways.

5 Conclusions and Future Work

In this paper, the initial steps toward the development of a novel anomaly detection model, inspired by T-cell recognition mechanisms, were presented. So far, the model incorporates MHC processing of antigens presented to the system, through the application of feature selection/extraction algorithms [53]. This leads to the generation of a number of peptides equals to a parameter analogous to the number of MHC isoforms, along with an antigen windowing approach, where the most recently analyzed antigens are retained in the system. Finally, an initial description of a biologically more plausible model of a T cell was presented, to be incorporated in the proposed approach, including multiple receptors and thus allowing these cells to receive several signals.

Future work will be conducted to the incorporation of tunable cellular thresholds, which, along with the definition of mechanisms for interactions between cells, will constitute the starting point to an implementation.

Acknowledgments. The authors would like to thank the financial support provided by UOL (www.uol.com.br, process number 20060519110414a), FAPEMIG and CNPq (process number 307178/2004-8).

References

1. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach, 1st edn. Springer, London (2002)
2. Markou, M., Singh, S.: Novelty detection: a review - part 1: statistical approaches. Signal Process 83(12), 2481–2497 (2003)
3. Markou, M., Singh, S.: Novelty detection: a review - part 2: neural network based approaches. Signal Process 83(12), 2499–2521 (2003)
4. Kim, J., Bentley, P.J., Aickelin, U., Greensmith, J., Tedesco, G., Twycross, J.: Immune system approaches to intrusion detection - a review. Nat. Comput. (in press) DOI: 10.1007/s11047-006-9026-4 XX (2007) XX–XX
5. Dasgupta, D., Yu, S., Majumdar, N.S.: MILA - multilevel immune learning algorithm and its application to anomaly detection. Soft Comput. 9, 172–184 (2005)
6. Grossman, Z., Paul, W.E.: Adaptive cellular interactions in the immune system: The tunable activation threshold and the significance of subthreshold responses. PNAS 89, 10365–10369 (1992)
7. Grossman, Z., Singer, A.: Tuning of activation thresholds explains flexibility in the selection and development of T cells in the thymus. PNAS 93, 14747–14752 (1996)
8. Grossman, Z., Paul, W.E.: Self-tolerance: context dependent tuning of T cell antigen recognition. Semin. Immunol. 12, 197–203 (2000)
9. Grossman, Z., Paul, W.E.: Autoreactivity, dynamic tuning and selectivity. Curr. Opin. Immunol. 13, 687–698 (2001)
10. Janeway, C.A., Travers, P., Walport, M., Shlonmchik, M.: Immunobiology: the immune system in health and disease, 5th edn. Garland Publishing, Inc. New York, USA (2002)
11. Germain, R.N., Stefanova, I.: The dynamics of T cell receptor signalling: Complex orchestration and the key roles of tempo and cooperation. Annu. Rev. Immunol. 17, 467–522 (1999)

12. Starr, T.K., Jameson, S.C., Hogquist, K.A.: Positive and negative selection of T cells. Annu. Rev. Immunol. 21, 139–176 (2003)
13. Depoil, D., Zaru, R., Guiraud, M., Chauveau, A., Harriague, J., Bismuth, G., Utzny, C., Müller, S., Valitutti, S.: Immunological synapses are versatile structures enabling selective T cell polarization. Immunity 22(2), 185–194 (2005)
14. Rachmilewitz, J., Lanzavecchia, A.: A temporal and spatial summation model for T-cell activation: signal integration and antigen decoding. Trends Immunol. 23(12), 592–595 (2002)
15. Friedl, P., Gunzer, M.: Interaction of T cells with APCs: the serial encounter model. Trends Immunol. 22(4), 187–191 (2001)
16. George, A.J.T., Stark, J., Chan, C.: Understanding specificity and sensitivity of T-cell recognition. Trends Immunol. 26(12), 653–659 (2005)
17. Davis, M.M., Krogsgaard, M., Huse, M., Huppa, J., Lillemeier, B.F., Li, Q.: T cells as a self-referential, sensory organ. Annu. Rev. Immunol. 25, 681–695 (2007)
18. Krogsgaard, M., Davis, M.M.: How T cells 'see' antigen. Nat. Immunol. 6, 239–245 (2005)
19. Valitutti, S., Müller, S., Cella, M., Padovan, E., Lanzavecchia, A.: Serial triggering of many T-cell receptors by a few peptide-MHC complexes. Nature 375, 148–151 (1995)
20. McKeithan, T.W.: Kinetic proofreading in T-cell receptor signal transduction. PNAS 92(11), 5042–5046 (1995)
21. van den Berg, H.A., Burroughs, N.J., Rand, D.A.: Quantifying the strength of ligand antagonism in TCR triggering. B Math. Biol. 64, 781–808 (2002)
22. Goldstein, B., Faeder, J.R., Hlavacek, W.S.: Mathematical and computational models of immune-receptor signalling. Nat. Rev. Immunol. 4, 445–456 (2004)
23. Burroughs, N.J., van der Merwe, P.A: Stochasticity and spatial heterogeneity in T-cell activation. Immunol Rev. 216, 69–80 (2007)
24. van den Berg, H.A., Rand, D.A.: Quantitative theories of T-cell responsiveness. Immunol. Rev. 216, 81–92 (2007)
25. Noest, A.J.: Designing lymphocyte functional structure for optimal signal detection: *Voilà*, T cells. J theor. Biol. 207, 195–216 (2000)
26. van den Berg, H.A., Rand, D.A., Burroughs, N.J.: A reliable and safe T cell repertoire based on low-affinity T cell receptors. J theor. Biol. 209(4), 465–486 (2001)
27. van den Berg, H.A., Rand, D.A.: Antigen presentation on MHC molecules as a diversity filter that enhances immune efficacy. J theor. Biol. 224, 249–267 (2003)
28. Sousa, J.: Modeling the antigen and cytokine receptors signalling processes and their propagation to lymphocyte population dynamics. PhD thesis, Universidade de Lisboa (2003)
29. Stefanova, I., Dorfman, J.R., Germain, R.N.: Self-recognition promotes the foreign antigen sensitivity of naive T lymphocytes. Nature 420, 429–434 (2002)
30. Stetter, M.: Dynamic functional tuning of nonlinear cortical networks. Phys. Rev. E 73(3 Pt 1), 031903 (2006)
31. Carneiro, J., Paixão, T., Milutinovic, D., Sousa, J., Leon, K., Gardner, R., Faro, J.: Immunological self-tolerance: Lessons from mathematical modelin. J Comp. Appl. Math. 184(1), 77–100 (2005)
32. van den Berg, H.A., Rand, D.A.: Dynamics of T cell activation threshold tuning. J theor. Biol. 228, 397–416 (2004)
33. Matzinger, P.: Tolerance, danger and the extended family. Annu. Rev. Immunol. 12, 991–1045 (1994)
34. Scherer, A., Noest, A., de Boer, R.J.: Activation-threshold tuning in an affinity model for the T-cell repertoire. Proc. R Soc. Lond B 271, 609–616 (2004)

35. Garrett, S.M.: How do we evaluate artificial immune systems? Evol. Comput. 13(2), 145–178 (2005)
36. Dasgupta, D.: Advances in artificial immune systems. IEEE Comp. Int. Mag. 1(4), 40–49 (2006)
37. Freitas, A.A., Timmis, J.: Revisiting the foundations of artificial immune systems for data mining. IEEE Trans. Evol. Comput. (in press) DOI: 10.1109 /TEVC.2006.884042 XXX (2006) XXX
38. Hofmeyr, S.A., Forrest, S.: Immunity by design: An artificial immune system. In: Proc. of the GECCO, vol. 2, pp. 1289–1296. Morgan Kaufmann, San Francisco (1999)
39. Alves, R.T., Delgado, M.R., Lopes, H.S., Freitas, A.A.: An artificial immune system for fuzzy-rule induction in data mining. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) Parallel Problem Solving from Nature - PPSN VIII. LNCS, vol. 3242, pp. 1011–1020. Springer, Heidelberg (2004)
40. Watkins, A., Timmis, J., Boggess, L.: Artificial immune recognition system (AIRS): An immune-inspired supervised learning algorithm. Genetic Prog. and Evol. Mach. 5(3), 291–317 (2004)
41. Timmis, J., Neal, M.: A resource limited artificial immune system for data analysis. Knowl. Based Sys. 14(3-4), 121–130 (2001)
42. Neal, M.: An artificial immune system for continuous analysis of time-varying data. In: Timmis, J., Bentley, P. (eds.) Proc. of the 1st ICARIS. 1, 76–85 (2002)
43. Galeano, J.C., Veloza-Suan, A., González, F.A.: A comparative analysis of artificial immune network models. In: Proc. of the GECCO, pp. 361–368 (2005)
44. Gonzalez, F., Dasgupta, D.: Anomaly detection using real-valued negative selection. Genetic Prog. and Evol. Mach. 4(4), 383–403 (2003)
45. 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)
46. Crook, N., Goh, W.J., Hawarat, M.: Pattern recall in networks of chaotic neurons. BioSystems 87, 267–274 (2007)
47. Belatreche, A., Maguire, L.P., McGinnity, M.: Advances in design and application of spiking neural networks. Soft Comput. 11, 239–248 (2007)
48. 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)
49. Aickelin, U., Bentley, P.J., Cayzer, S., Kim, J., Mcleon, 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)
50. Twycross, J., Aickelin, U.: Towards a conceptual framework for innate immunity. Lect. Notes Comput. Sc. 3627, 112–125 (2005)
51. 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)
52. 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)
53. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford Univ. Press, Oxford (1995)

Bankruptcy Prediction Using Artificial Immune Systems

Rohit Singh and Raghu Nandan Sengupta

Department of Industrial and Managment Engineering,
Indian Institute of Technology Kanpur
Kanpur – 208016, India
rohit123singh@rediffmail.com, raghus@iitk.ac.in

Abstract. In this paper we articulate the idea of utilizing Artificial Immune System (AIS) for the prediction of bankruptcy of companies. Our proposed AIS model considers the financial ratios as input parameters. The novelty of our algorithms is their hybrid nature, where we use modified Negative Selection, Positive Selection and the Clonal Selection Algorithms adopted from Human Immune System. Finally we compare our proposed models with a few existing statistical and mathematical sickness prediction methods.

Keywords: immune system, artificial immune system, positive selection, negative selection, clonal selection, accounting variables, bankruptcy, sickness prediction.

1 Introduction

The vertebrae Immune System (IS) is a highly complex system which is tuned to the problem of detecting and eliminating infections. The main task of the Immune System is to detect any foreign infectious element (antigen) and trigger an immune response to eliminate them. The immune system generates antibodies that detect and eliminate these antigens. This problem of detecting antigens is often described as a problem of *distinguishing between the "self" and the "non-self"*, where we describe a "self" to be synonymous to that cell that it is not harmful for the body, while a "non-self" is one which is harmful for the body and which should be destroyed [12, 13].

If one concentrates on the problem of bankruptcy prediction from a set of companies in a given environment, then sick companies can be considered as the antigens that need to be detected in the system. Many statistical models have been proposed which take into account some financial ratios or accounting variables whose values apparently demonstrate good predictive power. Most models take a linear combination of these variables to arrive at a “score” or a probability of bankruptcy in the near future, giving higher weightage to those ratios that are believed to possess higher discriminating power.

Due to the popularity of some financial ratios amongst analysts, they have widely been used for predicting the health of a company, with no proof of their superiority as an evaluating criterion. Professional attitudes and practice in this area are dominated by ‘conventional wisdom’ rather than scientific theory [11].

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 131–141, 2007.

Statistical models may not prove to be reliable under every circumstance. There is no single model which is equally reliable for all economic environments in which companies exist. A model prepared for sickness prediction of private companies may not accurately predict sickness of a public company [6]. Similarly a model prepared using data of companies in the United States may not accurately predict sickness of companies in India, the economic structure of these two countries being very different. Hence there is a need for a model which is flexible enough to incorporate environmental variability.

We propose a methodology for modeling an Artificial Immune System (AIS) for sickness prediction of a company. A set of accounting variables are used to represent a company. The values of these set of ratios provide a unique *signature* of a company – it is the property of the company and each company will have a different signature. This signature can be used to classify companies in the AIS context as either self or non-self.

We have selected a sample consisting of both sick and non-sick *Indian* companies, status known *a priori*. This is used for both training (generation of detectors), and validation of our model.

2 Statistical Methods in Sickness Prediction

The history of credit scoring model dates back to the seminal work of Altman [1], where the author uses Multivariate Discriminant Analysis to arrive at a linear combination of five financial ratios, called the Z-score, for predicting whether a company is credit worthy or not. Following Altman's [1] work, many different models have been proposed, like Altman *et al.* [2, 3], Ohlson [14], Zavgren [15], Zmijewski [16], and Griffin and Lemmon [10].

For all these models the underlying notion has been to use the different accounting and financial figures to arrive at a score or a probability of failure. Depending on the score or the probability of failure we arrive at a decision whether a particular company is doing well or not and whether it is credit worthy or not.

2.1 Altman's Z-Score

Altman [1] proposes a quantitative metric, called the Z-Score, for predicting the bankruptcy or sickness of industrial companies in USA. The author considers a sample of sixty-six corporations divided equally into two groups – bankrupt and non-bankrupt. Using twenty-two financial variables (which were important in prediction the financial performance of any corporate) for the period 1946-1965, Multivariate Discriminant Analysis was carried out to arrive at the Z-Score, the formula for which is:

$$Z = 1.2*X_1 + 1.4*X_2 + 3.3*X_3 + 0.6*X_4 + 0.999*X_5 \qquad (1)$$

where,
Z = Overall Index
X_1 = Working Capital/Total Assets
X_2 = Retained Earnings/Total Assets
X_3 = Earning before Interest and Income Tax/Total Assets

X_4 = Market value of Equity/Book value of Total Liabilities
X_5 = Sales/Total Assets

Apart from the above mentioned general formula for the Z-Score, variants of the score have also been modeled for two different types of companies, namely the public industrial companies and the private industrial companies.

2.2 ZETA^{TM}–Score[1]

The ZETA^{TM}-Score proposed in [2] is a modification of the Z-Score. Changes in accounting standards and government rules regarding bankruptcy, and focus on much larger firms led to this modification. The different variables which were considered to be important for calculating the ZETA^{TM}-Score are:

1. X_1 = Return on Assets (ROA) = EBIT/Total Sales
2. X_2 = Stability of Earnings. It measured the normalized standard error of estimate in X_1
3. X_3 = Debt Service = Interest Coverage Ratio = EBIT/Total Interest Payments
4. X_4 = Cumulative profitability = Retained Earnings. It gives a picture about the of the age of the firm, about the dividend policy of the firm and about the profitability of the firm over time
5. X_5 = Liquidity = Current Ratio = (Current Assets/Current Liabilities)
6. X_6 = Capitalization = (Common Equity/Total Capital)
7. X_7 = Size = $\log_e$(Total Assets)

2.3 O-Score

The prediction models developed till 1980 did not consider the probabilistic nature of operations of corporate, hence the element of uncertainty was absent in all the models. To incorporate the concept of probability in formulating the bankruptcy prediction scores Ohlson [14] put forward the following score, called the O-Score:

$$O = -1.32 - 0.407*Y_1 + 6.03*Y_2 - 1.43*Y_3 + 0.076*Y_4 - 2.37*Y_5 - 1.83*Y_6 + 0.285*Y_7 - 1.72*Y_8 - 0.521*Y_9 \quad (2)$$

where,
O = Overall Index used to calculate the probability of failure
Y_1 = log(Total Assets/GNP Price Index)
Y_2 = Total Liabilities/Total Assets
Y_3 = Working Capital/Total Assets
Y_4 = Current Liabilities/Current Assets
Y_5 = One if total liabilities exceeds total assets, zero otherwise
Y_6 = Net Income/Total Assets
Y_7 = Fund from Operations/Total Liabilities
Y_8 = One if net income was negative for the last two years, zero otherwise
Y_9 = ((Net Income (t) - Net Income (t-1)/|Net Income (t) - Net Income (t-1)|)
t = current year

[1] This is a proprietary model for subscribers to ZETA Services, Inc. (Hoboken, NJ), so the detailed model is not presented here.

2.4 Emerging Market (EM)-Score

Due to increasing globalization and more international trade and commerce, it became imperative to include the effects of countries, foreign currencies, industry characteristics, environment, political climate, economic climate, lack of credit experience in some economies etc., in formulating the sickness prediction scores. This resulted in the EM-Score model [3].

$$\text{EM-score} = 6.56(X_1) + 3.26(X_2) + 6.72(X_3) + 1.05(X_4) + 3.25 \tag{3}$$

where,
X_1 = working capital/total assets
X_2 = retained earnings/total assets
X_3 = operating income/total assets
X_4 = book value of equity/total liabilities

3 Data Collection

The sample consists of two groups of companies, viz. the bankrupt and the non-bankrupt manufacturing units. The bankrupt group consists of companies that file a petition under the Sick Industrial Companies (Special Provision) Act 1985 at the Board for Industrial & Financial Reconstruction (BIFR)[2] India. There was no restriction placed on size of the company to be selected for our dataset, but only those were considered for the experiment for which all the data required was available and consistent.

The data collected for all companies was taken from Prowess, Indian Corporate Database, provided by CMIE[3]. This database consists of the data for all those companies that are listed either on Bombay Stock Exchange (BSE)[4] or on National Stock Exchange (NSE)[5].

Data was collected over three years – 2002 to 2004. In our data set, the non-bankrupt group for each year consists of those companies which did not file a sickness petition between 2003 and 2006. The year of bankruptcy of a company was taken to be the year when it first filed the petition, and the data of bankrupt group was dated one financial year prior to the date of filing of bankruptcy (see Table 1).

Table 1. Description of data set

Year	Total No. of Companies	No. of companies in non-bankrupt group	No. of companies in bankrupt group
2002	126	94	32
2003	147	98	49
2004	144	121	23

[2] http://www.bifr.nic.in/
[3] http://www.cmie.com/
[4] http://www.bseindia.com/
[5] http://www.nseindia.com/

4 An Artificial Immune System for Sickness Prediction

An AIS system is designed for distinguishing between self and non-self entities. In the sickness prediction context, the *self* is defined as a financially healthy company, while the *non-self* is defined as a company that is sick. The AIS maintains a set of *detectors* for recognizing *non-self* companies.

In literature different algorithms can be found that have successfully been used to engineer AIS. Some of these are the *negative selection* [9], *positive selection* [7], and the *Clonal selection algorithm CLONALG* [8]. Some attempts have been made for the bankruptcy prediction and the bond rating of companies using the negative selection algorithm [4].

We have attempted to use the hybrid of these algorithms to engineer our Artificial Immune Sickness Prediction System. This is discussed below in detail.

4.1 Development of the Model

Antibodies, Antigens, and Detectors. The set of non-bankrupt companies in our training set serve as the self set, while the bankrupt set are treated as antigens/non-self. Using these two data, a set of antibodies/detectors is generated, which will be used for classifying the test data as either sick or non-sick.

In our study an antibody is represented as a string of *G* elements. Each element is a real-valued financial variable. These variables and the string length may be varied according to different economic scenarios. The same scheme is used for representing self and antigens.

Matching Function. The matching of a self or an antigen with the generated detector is done by calculating the Euclidean distance between them. It is done by using the following formula:

$$e = \frac{1}{\sqrt{G}} \sqrt{\sum_{g=1}^{G} \left(\frac{x_g - y_g}{range_g} \right)^2} \tag{4}$$

$$range_g = max_g - min_g \tag{5}$$

Where,
x = the self/non-self, represented as $\{x_1, x_2 \ldots x_g \ldots x_G\}$;
y = the detector, represented as $\{y_1, y_2 \ldots y_g \ldots y_G\}$;
G= number of elements in the string

Many other distance measures can be found in literature [5]. However, our model is fairly simple: it uses real-valued representation, there is no overlap between the elements of a string, and the data can be ordered according to their value. Hence, and for simplicity, the Euclidean distance measure has been used effectively in our model without introducing any unintended bias.

4.2 Algorithms and Procedures

The AIS algorithms that we use in our model build upon the *negative selection*, *positive selection* and the *Clonal selection algorithm CLONALG*. We have used these basic algorithms as the building blocks of our procedures.

We have divided our work into two phases viz. *training* of the system, i.e., generation of the detector set, and *monitoring* or classification of test data.

Our basic procedure (Procedure-1) for training uses the negative selection algorithm to generate the detector set, which will be used for classifying test data. We further refine the detector set using either positive selection or clonal selection (Procedure-2 and Procedure-3). Finally, we compare the performance of these three procedures.

The negative selection algorithm [9] (Table 2) takes as input a set of *self*-strings that define the set of healthy companies in our application, and generates *n detectors* that are not capable of recognizing any self-string. To achieve this, random strings are generated and matched with each self-string to get the Euclidean distance. If a mismatch occurs, i.e. the distance when matched with *all* self-strings is greater than the cross reactivity threshold, *r*, the random string is taken to be a detector.

The positive selection algorithm [7] (Table 3) takes as input a set of strings M, and matches them against a set of *non-self* strings, *NS*. If a match occurs, i.e. the distance when matched with *any* self-string is lesser than the cross reactivity threshold, *r*, the random string is selected for the optimized set *A*. The rest of the strings in *M* are rejected.

Table 2. Negative Selection Algorithm

```
Algorithm: Negative Selection
Input: self set S, cross reactivity threshold r, no. of detectors required
         n, and length of string L
Output: Detector Set A
Begin
         j ← 0
            While j <= n do
                  m ← random(1, L)
                  for each s of S do
                           dis ← match (m, s);
                           if dis >= r then
                                    insert (A, m)
                                    breakFor
                           endif
                  endFor
                  j ← j+1
         endWhile
         return A
end
```

Table 3. Positive Selection Algorithm

Algorithm: Positive Selection

Inputs: set of non-self strings *NS,* cross-reactivity threshold r_2, string-set *M* to be optimized
Outputs: optimized set *A*

```
Begin
    for each m of M do
        for each ns of NS do
            dist ← match (m, ns);
            if dist <= r_2 then
                insert (A,m)
                breakFor
            endif
        endFor
    endFor
    return A
end
```

Procedure 1: This procedure is *based* on the basic negative selection algorithm alone for generating the detector set. The self-set is the set of non-bankrupt companies for a given year and each detector has the property that it is unable to detect *at least* one self-string within the cross-reactivity threshold.

Procedure 2: Here we use a hybrid of the negative selection and positive selection algorithms. In this procedure, the detector-set generated by the negative selection algorithms is further refined by ensuring that each detector is able to detect at least one non-self-string used for training the system.

Procedure 3: Similarly, a hybrid of negative selection followed by clonal selection algorithm can also be used to refine the detector set.

5 Experiments

The above-mentioned procedures were tested using simulations on Matlab 7.0.1. For representing each company, we have taken nine financial ratios which have been shown to represent the state of a company with high fidelity [11, 16]. These ratios are the most commonly used balance sheet ratios used in current bankruptcy prediction systems. The values of these ratios are normalized between 0 and 100 to increase the precision while matching.

The data collected was partitioned into two – the training set and the test set. The details of the data set, parameters and experimental results are discussed below.

5.1 Test Sets

We used 30 companies from the non-bankrupt group and 15 from our bankrupt group as the self-set and non-self-set respectively for training our model for each year. The

Table 4. Test sets

Year	Total No. of Companies	No. of companies in non-bankrupt group	No. of companies in bankrupt group
2002	96	64	32
2003	127	68	49
2004	124	91	23

remaining non-bankrupt companies and all of the bankrupt companies were used as the test set. The characteristics of the test set are described in Table 4.

5.2 Parameters

There are three parameters that must be chosen – the number of detectors in the detector set, *n*, the cross-reactivity threshold for negative selection, r_1, and the cross-reactivity threshold for positive selection, r_2.

We have chosen *n* to be 100 for procedure-1 and 300 for procedure-2. The rationale behind taking a larger number for procedure-2 is that the optimized set after positive selection algorithm contains less than half the original number of detectors.

The r_1 and r_2 values must vary between 0 and 100 since the values of each string element is normalized on this scale. For choosing the optimum value, we conducted exhaustive tests for each possible combination of these two parameters on the 2004 data-set. For each test we computed the Type-I (8) and Type-II (9) errors for the following hypothesis:

$$H_0 = \text{All the companies detected by the detector set are bankrupt} \quad (6)$$

$$H_1 = \text{No companies detected by the detector set are bankrupt} \quad (7)$$

$$\begin{array}{c}\text{Type-I}\\ \text{Error}\\ \text{percentage}\end{array} = \frac{\text{number of bankrupt companies not classified as bankrupt}}{\text{total number of bankrupt companies}} \quad (8)$$

$$\begin{array}{c}\text{Type-II}\\ \text{Error}\\ \text{percentage}\end{array} = \frac{\text{number of non-bankrupt companies classified as bankrupt}}{\text{total number of non-bankrupt companies}} \quad (9)$$

We chose five r_1 - r_2 combinations for which the average sum of Type-I and Type-II error was the minimum for further analysis.

5.3 Results and Discussions

For each of the five r_1 - r_2 combinations chosen above, we performed 20 sets of 1000 runs each for the three data sets corresponding to the three years. These experimental runs were done for both procedures-1 and -2. We again computed the Type-I and

Type-II errors, as mentioned above, and found the average and the standard deviation of the sum of these errors for the 20 sets, averaged over the 1000 results.

Our results shown in Table 5 & 6 are for the best r_1 - r_2 combination, that came out to be r_1=47 and r_2=30.

Table 5. Classification results for procedure-1 with r_1=47 and r_2=30

Year	Type-I Error (%)		Type-II Error (%)	
	average	St. error	average	St. error
2002	0.01890	0.00845	39.8175	0.47202
2003	11.3277	0.33707	1.75147	0.02333
2004	3.8786	0.25307	2.71593	0.04419

Table 6. Classification results for procedure-2 with r_1=47 and r_2=30

Year	Type-I Error (%)		Type-II Error (%)	
	average	St. error	average	St. error
2002	0	0	25.1207	0.221
2003	5.4432	0.0902	1.5318	0.0099
2004	0.0943	0.028	3.0158	0.03788

Upon examination of the above results, we can conclude that the results obtained through procedure-2 exhibit higher accuracy rates than those obtained by procedure-1, consistently for all the years. Thus we can claim that by using positive selection for improving the detector set, the overall classification accuracy can be enhanced.

We observe a marked difference between the accuracy achieved in 2003 and 2004, and that achieved in 2002. This difference can be attributed to the fact that the non-bankrupt groups for 2003 and 2004 did not contain any company that filed for bankruptcy over a span of three years – i.e. the non-bankrupt group for a particular year contains companies that had maintained a good record for at least three consecutive years. This was not possible for the year 2002 due to unavailability of data for the years 2000 and 2001. In spite of this shortcoming, there is a significant improvement in the accuracy level of the results obtained for year 2002 through using procedure-2 over those obtained by using procedure-1.

5.4 Comparison with Other Sickness Prediction Models

To compare our test results obtained above we classified our data set using three statistical methods viz. the Altman Z-score , the Emerging Market score, and the Ohlson Score, and then calculated the errors for the classification results obtained.

It can be seen from Table 7 that for the year 2002 the accuracy of our AIS classification is far better than any of the results obtained by both the statistical methods. For the other two years our Type-1 error percentage is slightly higher than that of the Z-score. As far as the Type-II error rate for all the three years is concerned, both our proposed models always yield far better results than the Z-score model.

Upon comparison of the result of our AIS models with the EM-score results, we can see that although we have obtained a slightly higher Type-II error, we obtain a marked improvement in Type-I error rate.

Table 7. Classification results obtained by statistical models

Year	Z-score		EM-score	
	Type-I Error (%)	Type-II Error (%)	Type-I Error (%)	Type-II Error (%)
2002	9.375	36.56	53.125	0
2003	4	39.79	32.653	0
2004	0	43.8	30.434	0

The O-score results are not shown as it classified all the companies in our dataset as bankrupt giving a probability of bankruptcy for each one of them to be greater than 0.5.

The high error obtained in classification by the statistical methods can be attributed to the fact that these methods have constraints on the size, market and the economic environment of the companies which are not imposed in selection of our dataset.

6 Conclusion and Future Scope

In this paper, we explored the possibility of using the Artificial Immune Systems framework for predicting the sickness of a company over a period of one year. We have compared two different procedures using the basic negative selection and positive selection algorithms; and our results clearly show the advantage of using positive selection algorithms to optimize the detector set generated by the negative selection. However, both our techniques demonstrate very high accuracy.

Our method uses only two parameters, r_1 and r_2 which can be determined easily after a few simulations and statistical tests. The results, however exhibit high accuracy over a range of combinations of these parameters, regardless of the data set used for training. This offers us the opportunity of using different data sets for training depending on economic environment of the companies that we would like to classify. There is also a possibility of using more advanced distance measures or data-mining techniques for improving the detector set.

We also compared our results with classification results obtained by statistical methods and noted a marked difference in performance on the three data sets we have used.

These models can also be used for Credit Rating with slight modifications in data representation and can be easily designed to suit different economic environments by incorporating some macro-economic variables in the antigen/antibody representation.

There is also scope for finding the optimal set of financial variables to be used in our model which may exhibit enhanced results.

References

1. Altman, E.I.: Financial Ratios, Discriminant Analysis and Prediction of Corporate Bankruptcy. Journal of Finance. 23, 189–209 (1968)
2. Altman, E.I., Haldeman, R.G., Narayanan, P.: ZETA^{TM} Analysis: A New Model to identify Bankruptcy Risk of Corporations. Journal of Banking and Finance. 1, 29–54 (1977)
3. Altman, E.I., Hatzell, J., Peck, M.: Emerging Markets Corporate Bonds: A Scoring System. Salomon Brothers, New York, USA (1997)
4. Brabazon, A., O'Neill, M.: Biologically Inspired Algorithms for Financial Modelling (Natural Computing Series). Springer, New York (2006)
5. Boggess, L., Hamaker, J.S.: Non-Euclidean Distance Measures in AIRS, an Artificial Immune Classification System. Proceedings of the IEEE Congress on Evolutionary Computation. 1, 1067–1073 (2004)
6. Caouette, J.B., Altman, E.I., Narayanan, P.: Managing credit risk. Wiley Frontiers in Finance. John Wiley & sons Inc. New York (1998)
7. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, London (2002)
8. de Castro, L.N., von Zuben, F.J.: The Clonal Selection Algorithm with Engineering Applications. In: Proceedings of GECCO'00, pp. 36–37 (2000)
9. Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R.: Self-Nonself Discrimination in a Computer. In: Proceedings of the IEEE Symposium on Research in Security and Privacy, pp. 202–212. IEEE Computer Society Press, Los Alamitos, CA (1994)
10. Griffin, J.M., Lemmon, M.L.: Book to Market Equity, Distress Risk and Stock Returns. Journal of Finance. 57, 2317–2336 (2002)
11. Gupta, L.C.: Financial Ratios for Monitoring Corporate Sickness: Towards a More Systematic Approach. Oxford University Press, Delhi (1983)
12. Matzinger, P.: Tolerance, Danger, and the Extended Family. Annual Reviews of Immunology 12, 991–1045 (1994)
13. Matzinger, P.: An Innate Sense of Danger. Seminars in Immunology 10, 399–415 (1998)
14. Ohlson, J.: Financial Ratios and the Probabilistic Prediction of Bankruptcy. Journal of Accounting Research 18, 109–131 (1980)
15. Zavgren, C.V.: The Prediction of Corporate Failure, the State of the Art. Journal of Accounting Literature 2, 1–33 (1983)
16. Zmijewski, M.E.: Methodological Issues Related to the Estimation of Financial Distress Prediction Models. Journal of Accounting Research 22, 59–82 (1984)

Phase Transition and the Computational Complexity of Generating r-Contiguous Detectors

Thomas Stibor

Department of Computer Science
Darmstadt University of Technology
stibor@sec.informatik.tu-darmstadt.de

Abstract. The problem of generating r-contiguous detectors in negative selection can be transformed in the problem of finding assignment sets for a Boolean formula in k-CNF. Knowing this crucial fact enables us to explore the computational complexity and the feasibility of finding detectors with respect to the number of self bit strings $|\mathcal{S}|$, the bit string length l and matching length r. It turns out that finding detectors is hardest in the phase transition region, which is characterized by certain combinations of parameters $|\mathcal{S}|, l$ and r. This insight is derived by investigating the r-contiguous matching probability in a random search approach and by using the equivalent k-CNF problem formulation.

1 Introduction

Theoretical immunologists have proposed the r-contiguous matching function to abstract the affinity between an antibody and an antigen in immune system models [1]. In the field of artificial immune systems, the r-contiguous matching function is applied as a matching rule for change detection [2] or more generally for anomaly detection problems. In these domains, antibodies (called detectors) and antigens are abstracted as bit strings and the r-contiguous matching rule is applied for detecting (anomalous) antigens. More specifically, in this immune inspired anomaly detection approach, the problem is to find detectors, such that no detector match with any self antigen. This form of detector generation for the complementary space is called *negative selection* [3].

In recent years, many attempts were made (see [4,5] for an overview) to generate detectors efficiently, i.e. in polynomial time and with polynomial space occupation with regard to the matching length r and number of self antigens $|\mathcal{S}|$. All attempts in designing efficient algorithms for generating r-contiguous detectors were limited successful. The proposed algorithms either have a time or a space complexity which is exponential[1] in the matching length r, i.e. $\mathcal{O}(2^r)$ or in the number of self elements $|\mathcal{S}|$, i.e. $\mathcal{O}(e^{|\mathcal{S}|})$. Stibor et al. [6] proved that the

[1] There exists a *linear time detector generating algorithm* [2], however this algorithm still requires $\mathcal{O}(2^r)$ time and space occupation. It is termed linear, because it runs linear in $|\mathcal{S}|$ under the *assumption* that $|\mathcal{S}| = \mathcal{O}(2^r)$. For real-world problems however, the assumption $|\mathcal{S}| \ll 2^r$ is justifiable.

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 142–155, 2007.

problem of generating r-contiguous detectors can be transformed in a k-CNF satisfiability problem and argued that at least $\Omega(2^r)$ bit string evaluations are required to find *all* r-contiguous detectors.

In this paper we go one step further and explore the computational complexity of generating detectors with the Davis-Logemann-Loveland algorithm. Furthermore, we rigorously analyze, when detectors can be generated with respect to the number of self bit strings $|\mathcal{S}|$, the bit string length l and matching length r. It will turn out that generating r-contiguous detectors is computationally *not* equally "hard". More specifically, it is relatively cheap computationally, to verify that *no* detectors can be generated or that a large number of detectors can be generated. However, there also exists a *phase transition* region which is characterized by certain combinations of parameters $|\mathcal{S}|, l$ and r where finding detectors is hardest. This insight will be derived from two directions, namely by investigating the r-contiguous matching probability in a random search approach and by using the problem transformation of generating detectors into the k-CNF satisfiability problem.

2 Bit String Matching Rule and Generating Detectors Randomly

Let $\mathcal{U}$ be a universe which contains all 2^l distinct bit strings of length l.

Definition 1. *A bit string $b \in \mathcal{U}$ with $b = b_1 b_2 \ldots b_l$ and detector $d \in \mathcal{U}$ with $d = d_1 d_2 \ldots d_l$, match with r-contiguous rule, if a position p exists where $b_i = d_i$ for $i = p, \ldots, p + r - 1$ and $p \leq l - r + 1$.*

Loosely speaking, two bit strings, with the same length, match if at least r contiguous bits are identical. In the remaining sections the expression "detectors" will refer to r-contiguous detectors. Sets are denoted in calligraphic letters, e.g. $\mathcal{S}$ and $|\mathcal{S}|$ denotes the cardinality. Throughout the paper, we will assume that $\mathcal{S}$ contains pairwise distinct bit strings randomly drawn from $\mathcal{U}$.

2.1 Randomly Generating Detectors in Negative Selection

Given $\mathcal{U}$ and its partition into distinct subsets $\mathcal{S}$ and $\mathcal{N}$. In negative selection one has to find detectors such that no detector matches (see Def. 1) with any bit string from $\mathcal{S}$. Detectors which satisfy this property match with — not necessarily all — bit strings from the complementary space $\mathcal{U} \setminus \mathcal{S}$. Algorithm (1) is a straightforward random search to generate, i.e. to find detectors. A bit string d is randomly sampled from $\mathcal{U}$ and matched against all bit strings in $\mathcal{S}$. When no r-contiguous match occurs, d is added to the detector set $\mathcal{D}$. This random sampling is repeated until a certain number of detectors is found. It is obvious that this straightforward random search is not an efficient search technique. However, a thorough probabilistic analysis of algorithm (1) reveals valuable insights, whether detectors can or can not be generated.

Algorithm 1. Random search for detectors in negative selection

```
input  : $l, r, t \in \mathbb{N}$ where $1 \leq r \leq l$ and $\mathcal{S} \subset \mathcal{U}$
output: Set $\mathcal{D} \subset \mathcal{U}$ of $r$-contiguous detectors
begin
    $\mathcal{D} := \emptyset$
    while $|\mathcal{D}| < t$ do
        Sample randomly a bit string $d \in \mathcal{U}$
        if $d$ does not match with any bit string of $\mathcal{S}$ then
            $\mathcal{D} := \mathcal{D} \cup \{d\}$
end
```

2.2 Probability of Matching in Random Detector Generation

The probability that two randomly drawn bit strings from $\mathcal{U}$ are *not* matching with the r-contiguous rule can be determined with approaches from probability theory, namely recurrent events and renewal theory [7]. In Feller's textbook [7] on probability theory an equivalent[2] problem is formulated as follows:

> "*A sequence of n letters S and F contains as many S-runs of length r as there are non-overlapping uninterrupted blocks containing exactly r letters S each*".

Given a Bernoulli trial with outcomes S (success) and F (failure), the probability of no success running of length r in l trials is according to Feller

$$P_{WF} = \frac{1 - px}{(r + 1 - rx)q} \cdot \frac{1}{x^{l+1}} \tag{1}$$

where

$$p = q = \frac{1}{2} \quad \text{and} \quad x = 1 + qp^r + (r+1)(qp^r)^2 + \ldots$$

A simpler approximation — however only valid for $r \geq l/2$ — is provided by Percus et al. [1]:

$$P_{JP} = 1 - 2^{-r}\left[(l-r)/2 + 1\right]. \tag{2}$$

From (1) one can straightforwardly conclude that the probability of finding t detectors when given l, r and $|\mathcal{S}|$ results in:

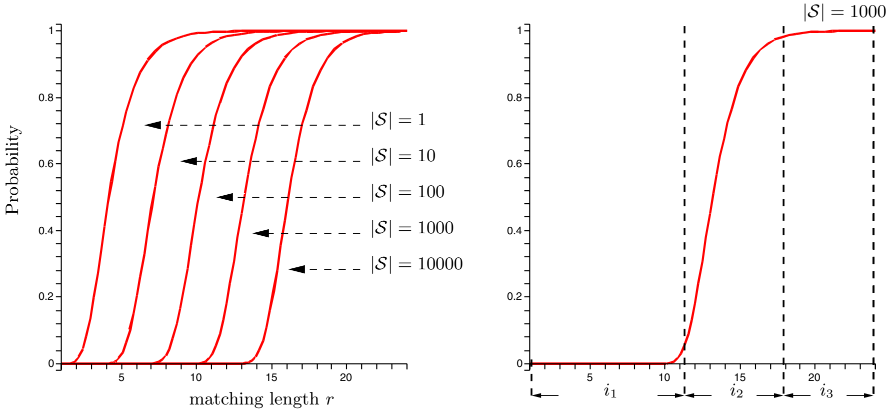
$$\textbf{Prob}[\text{find } t \text{ detectors}] = t^{-1} \cdot (P_{WF})^{|\mathcal{S}|}. \tag{3}$$

[2] The Link between recurrent events, renewal theory and the r-contiguous matching rule was discovered originally by Percus et al. [1] and rediscovered by Ranang [8]. Percus et al. presented in [1] the approximation (2) which is only valid for $r \geq l/2$, but mentioned the full approximation for $1 \leq r \leq l$ indirectly by mentioning the name de Moivre and citing Uspensky's textbook (see pp. 77 in [9]).

Moreover, from (3) one can conclude how often on average step 4 in algorithm (1) is executed when given t, or in other words how many bit strings one has to sample before finding $\widehat{t}$ detectors.

$$\widehat{t} = \frac{1}{t^{-1} \cdot P_{WF}^{|\mathcal{S}|}}. \tag{4}$$

Result (4) is equivalent to an earlier result on negative selection [3], when P_{WF} is replaced by P_{JP}.



(a) Matching probability for finding a detector randomly for $l := 24$, $r := [1 \dots 24]$ and $|\mathcal{S}| := \{1, 10, 100, 1000, 10000\}$.

(b) If r lies within interval i_1, then with high probability no detectors will be found, whereas if r lies within interval i_3, then with high probability, detectors will be found. There also exists an interval i_2 where the probability rapidly changes from 0 to 1.

Fig. 1. Coherence between the probability of finding a detector randomly and the parameters l, r and $|\mathcal{S}|$. There exists a sharp transition boundary where the probability rapidly changes from 0 to 1.

2.3 Probability Transition in r-Contiguous Matching

Knowing the probability P_{WF} enables us to investigate the combinations of parameters $|\mathcal{S}|, l$ and r where, with high probability detectors can be generated (i.e. exist) or with high probability can not be generated. The graphs in figure 1 show the probability for finding a detector for fixed l and variable r and $|\mathcal{S}|$ according to term (3). One can see, that the larger the cardinality of $\mathcal{S}$, the larger the interval for r where the resulting probability is nearly 0 to find a detector, or in other words where no detectors exist. On the other hand, the smaller the cardinality of $\mathcal{S}$, the larger the interval for r where the resulting probability is nearly 1 to find a detector. In figure 1(b) the same graph, but only for $|\mathcal{S}| = 1000$ is highlighted. One can see in detail that three different intervals (i_1, i_2, i_3) exist. One can either find with high probability a detector in interval i_1, or find with high probability no detector in interval i_3. Moreover there exists a third interval i_2 where the probability rapidly changes from 0 to 1. For the sake of conformity with the subsequent sections, we denote the interval i_2 as *phase transition* region. We will later see, that finding detectors in this region, which is characterized by certain combinations of parametes $|\mathcal{S}|, l, r$ is hardest from the perspective of computationally complexity.

To summarize this section, if parameters $|\mathcal{S}|, l$ and r are chosen such that term (3) results in a value very close to 0, then in the worst case no detectors can be generated, never mind which algorithms, i.e. search techniques are applied to generate detectors, because there exist no detectors. On the other hand, if term (3) is close to 1, then a large number of detectors exist.

2.4 Coherence of Matching Length r, Self Set $\mathcal{S}$ and Random Detector Search

In the artificial immune system community seems to exist some confusion regarding the time complexity of algorithm (1). More specifically, authors in [3] argued that generating detectors when applying the random search approach can be performed linearly in $|\mathcal{S}|$. Their argument is based on the observation that $\widehat{t}$ in (4) is minimized by choosing $1 - P_{JP} \approx 1/|\mathcal{S}|$. In other words, the number of bit strings one has to sample before finding t detectors is linear proportionally to $|\mathcal{S}|$, when using algorithm (1). This observation implies that the matching threshold r purely depends on the cardinality of S when l is fixed. To be more precise, suppose $r \geq l/2$, then

$$2^{-r}\left[(l-r)/2+1\right] \approx |S|^{-1} \Longleftrightarrow r \approx l + 2 - \frac{W(8\ln(2)2^l/|S|)}{\ln(2)} \qquad (5)$$

where $W(x)$ is the Lambert W-function which can be expressed as the series expansion

$$W(x) = \sum_{k=1}^{\infty} \frac{(-1)^{k-1}k^{k-2}}{(k-1)!}x^k. \qquad (6)$$

Practically speaking, once $|S|$ and l are fixed, the matching length r is such chosen that it will fall in interval i_3 (see Fig. 1(b)) and consequently this implies that a large number of detectors can be generated.

With regards to anomaly detection problems, it is known [10,11,12] that the r-contiguous matching rule is a positional biased detection rule. That means that the value of r is inextricably linked to the underlying data being analyzed. The assumption $1 - P_{JP} \approx 1/|\mathcal{S}|$ however, implies that r grows with $|\mathcal{S}|$ (see term (5)), and does not consider the positional bias. On the other side, if l and r are fixed[3] and $|\mathcal{S}|$ is considered as the variable parameter then $\widehat{t} = \mathcal{O}(e^{|\mathcal{S}|})$, that

[3] To capture the semantical representation of the data being analyzed.

is, r will lie within interval i_1 for some large $|\mathcal{S}|$ and this consequently implies that a random search for detectors results in an exponential time complexity — when detectors exist.

2.5 Average Number of Detectors and Holes

For the sake of completeness, we present results on the average number of detectors that can be generated and the resulting holes. The results are straightforward conclusions from the previous section 2.2.

Recall, algorithm (1) fails to find any detector when a certain parameter combination of $\mathcal{S}, l$ and r exists. More specifically, the universe $\mathcal{U}$ is not only composed of sets $\mathcal{S}, \mathcal{D}$ and $\mathcal{N}$, but also of set $\mathcal{H}$. Recall, the set $\mathcal{N}$ contains all bit strings which are detectable by the detectors from $\mathcal{D}$ and hence $\mathcal{D} \subset \mathcal{N}$. The set $\mathcal{H}$, termed hole set contains all bit strings which are not detectable by any detector, however, $\mathcal{H}$ does not contain any bit strings from $\mathcal{S}$, i.e. $\mathcal{H} \cap \mathcal{S} = \emptyset$ and hence, $|\mathcal{H}|$ is directly linked with interval i_1 (see Fig.1(b)). More specifically, if a parameter combination of l, r and $\mathcal{S}$ is chosen such that term (3) is very close to 0, then $|\mathcal{N}| \ll |\mathcal{H}|$ or in the extreme case $|\mathcal{N}| = 0$, i.e. the universe $\mathcal{U}$ is only composed of sets $\mathcal{S}$ and $\mathcal{H}$.

Knowing this coherence between term (3) and the universe composition, the average number of detectors that can be generated results in

$$\mathbf{E}[|\mathcal{D}|] = 2^l \cdot (P_{WF})^{|\mathcal{S}|}. \tag{7}$$

As the universe is composed of $\mathcal{U} = \mathcal{S} \cup \mathcal{N} \cup \mathcal{H}$ when applying the negative selection, the number of holes results in

$$|\mathcal{H}| = |\mathcal{U}| - |\mathcal{N}| - |\mathcal{S}| \tag{8}$$

where

$$\mathbf{E}[|\mathcal{N}|] = 2^l \quad - \quad \underbrace{2^l \cdot (P_{WF})^{\mathbf{E}[|\mathcal{D}|]}}_{\substack{\text{Number of bit strings} \\ \text{not detected by } \mathbf{E}[|\mathcal{D}|] \\ \text{detectors}}} \tag{9}$$

and hence the average number of holes results in

$$\mathbf{E}[|\mathcal{H}|] = 2^l \cdot (P_{WF})^{\mathbf{E}[|\mathcal{D}|]} - |\mathcal{S}|. \tag{10}$$

3 Link Between *r*-Contiguous Detectors and *k*-CNF Satisfiability

Stibor et al. [6] proved that the problem of generating detectors in negative selection can be transformed in an equivalent problem of finding assignment sets for a Boolean formula in k-CNF. Satisfying a Boolean formula in k-CNF is an instance of the satisfiability problem [13], where one has to decide if there is some assignment of *true* and *false* values that will make a Boolean formula in conjunctive normal form *true*. For the sake of clarity, we summarize the transformation steps presented in [6].

Let $b \in \{0,1\}$ and $\mathfrak{L}(b)$ a mapping defined as:

$$\mathfrak{L}(b) \rightarrow \begin{cases} x & \text{if} \quad b = 0 \\ \overline{x} & \text{otherwise} \end{cases}$$

where $x, \overline{x}$ are literals. Moreover, let $k, l \in \mathbb{N}$, where $k \leq l$ and $s \in \mathcal{U}$, where $s[i]$ denotes the bit at position i of bit string s. A mapping from bit string s into the l-k-CNF[4] is defined as follows:

$$\begin{aligned} \mathfrak{C}(s,k) \rightarrow\ & (\mathfrak{L}(s[1]) \ \vee\ \mathfrak{L}(s[2]) \ \vee\ \ldots\ \vee\ \mathfrak{L}(s[k])) \ \wedge \\ & (\mathfrak{L}(s[2]) \ \vee\ \mathfrak{L}(s[3]) \ \vee\ \ldots\ \vee\ \mathfrak{L}(s[k+1])) \ \wedge \\ & \qquad\qquad\qquad \vdots \\ & (\mathfrak{L}(s[l-k+1]) \ \vee\ \ldots\ \vee\ \mathfrak{L}(s[l]))\,. \end{aligned}$$

The resulting Boolean formula is constructed by an AND-combination of all bit strings in $\mathcal{S}$, i.e.

$$\widehat{\phi}_{rcb} := \mathfrak{C}(s_1,k) \ \wedge\ \mathfrak{C}(s_2,k) \ \wedge\ \ldots\ \wedge\ \mathfrak{C}(s_{|\mathcal{S}|},k) \text{ for } s_i \in \mathcal{S},\ i = 1,\ldots,|\mathcal{S}|$$

Proposition 1 (Stibor et al. [6]). *Given a universe $\mathcal{U}$ which contains all 2^l distinct bit strings of length l, a set $\mathcal{S} \subset \mathcal{U}$ and the set $\mathcal{D}$ which contains all generable r-contiguous detectors, which do not match any bit string from $\mathcal{S}$. The Boolean formula $\widehat{\phi}_{rcb}$ which is obtained by $\mathfrak{C}(s,r)$ for all $s \in S$ is satisfiable only with the assignment set $\mathcal{D}$.*

To summarize, instead of searching for detectors e.g. by means of algorithm (1), one can use SAT-Solvers [14] to find assignments sets of $\widehat{\phi}_{rcb}$. This crucial fact can be exploited for quantifying the computational complexity of finding detectors. However, one must estimate the *average number of distinct clauses* after applying the transformation steps, otherwise one would consider equal clauses several times — and this consequently would make the problem "harder" then it is.

3.1 Average Number of Distinct Clauses

Let $\mathcal{S}$ be a subset of $\mathcal{U}$ which contains pairwise distinct bit strings $s_1, s_2, \ldots, s_{|\mathcal{S}|}$ which are randomly drawn from $\mathcal{U}$. The constructed Boolean formula $\widehat{\phi}_{rcb}$ does not necessarily contains pairwise distinct clauses. Two clauses are distinct from each other, if they differ in at least one literal.

[4] The Boolean formula is denoted as l-k-CNF, because it is a special type of a k-CNF.

Example 1. Let $\mathcal{S} := \{0101, 1101\}$ and $r = 3$, hence $\widehat{\phi}_{rcb}$ results in

$$(x_1 \vee \overline{x}_2 \vee x_3) \wedge (\overline{x}_2 \vee x_3 \vee \overline{x}_4) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee x_3) \wedge (\overline{x}_2 \vee x_3 \vee \overline{x}_4).$$

Example 1 shows that the second and the fourth clause are equal, because the last three bits of 0101 and 1101 are equal.

Proposition 2. *Given bit string length l, matching length r and let $\mathcal{S}$ be a subset of $\mathcal{U}$ which contains pairwise distinct bit strings $s_1, s_2, \ldots, s_{|\mathcal{S}|}$ randomly drawn from $\mathcal{U}$. The average number of pairwise distinct clauses is*

$$\mathbf{E}[|\widehat{\phi}_{rcb}|] = 2^r\,(l-r+1) - \left(1 - \frac{1}{(l-r+1)\,2^r}\right)^{|\mathcal{S}|(l-r+1)} (l-r+1)\,2^r. \quad (11)$$

Proof. Construct a lookup table $\mathfrak{T}$ which contains all $2^r \cdot (l-r+1)$ clauses with label T and is of the form

clause	*label*
$(x_1 \vee x_2 \vee \ldots \vee x_{r-1} \vee x_r)$	T
$(x_2 \vee x_3 \vee \ldots \vee x_r \vee x_{r+1})$	T
$\vdots$	$\vdots$
$(x_{l-r+1} \vee x_{l-r+2} \vee \ldots \vee x_{l-1} \vee x_l)$	T
$(x_1 \vee x_2 \vee \ldots \vee x_{r-1} \vee \overline{x}_r)$	T
$(x_2 \vee x_3 \vee \ldots \vee x_r \vee \vee \overline{x}_{r+1})$	T
$\vdots$	$\vdots$
$(x_{l-r+1} \vee x_{l-r+2} \vee \ldots \vee x_{l-1} \vee \overline{x}_l)$	T
$\vdots$	$\vdots$
$(\overline{x}_1 \vee \overline{x}_2 \vee \ldots \vee \overline{x}_{r-1} \vee \overline{x}_r)$	T
$(\overline{x}_2 \vee \overline{x}_3 \vee \ldots \vee \overline{x}_r \vee \overline{x}_{r+1})$	T
$\vdots$	$\vdots$
$(\overline{x}_{l-r+1} \vee \overline{x}_{l-r+2} \vee \ldots \vee \overline{x}_{l-1} \vee \overline{x}_l)$	T

Transform $\mathcal{S}$ into the corresponding Boolean formula $\widehat{\phi}_{rcb}$ and set the label to F whenever a clause in $\mathfrak{T}$ is member of $\widehat{\phi}_{rcb}$. As $\mathcal{S}$ is randomly drawn without replacement from $\mathcal{U}$, the F and T labels are binomially distributed in $\mathfrak{T}$. The probability of finding *no* clauses which are labeled with F when randomly drawn $|\mathcal{S}| \cdot (l-r+1)$ clauses from $\mathfrak{T}$ results in

$$\left(1 - \frac{1}{(l-r+1)\,2^r}\right)^{|\mathcal{S}|(l-r+1)}$$

and hence, the total number of clauses with label F results in

$$2^r\,(l-r+1) - \left(1 - \frac{1}{(l-r+1)\,2^r}\right)^{|\mathcal{S}|(l-r+1)} (l-r+1)\,2^r. \qquad \square$$

4 Computational Complexity of Generating Detectors

A common approach to quantify the computational "hardness" of an instance of a Boolean formula in k-CNF is to count the number of backtracking attempts in the Davis-Logemann-Loveland (DLL[5]) algorithm. The DLL algorithm [17] is based on the elimination rules proposed by Davis and Putnam [18] and terminates either with result unsatisfiable (empty clause) or satisfiable (empty ϕ). More specifically, the algorithm is a depth-first search technique and uses recursive backtracking for guiding the exploration. The algorithm constructs a decision tree, where assignments of the variables coincide with paths from the root to the leafs. If a path leads to an unsatisfiable result, then the algorithm backs up to a different branch. This recursive search is reiterated until it terminates with a satisfiable or unsatisfiable result. In the worst case the whole decision tree has to be inspected, i.e. it will take an exponential number of evaluations — similar to an exhaustive search. However on average the DLL algorithm is much faster because it can prune whole branches from the decision tree without exploring the leaves.

Given a Boolean formula ϕ in CNF, a literal l in ϕ and the reduction function $R(\phi, l)$ that outputs the residual formula of ϕ by:

– removing all the clauses that contain l,
– deleting $\overline{l}$ from all the clauses that contain $\overline{l}$,
– removing both l and $\overline{l}$ from the list of literals.

A clause that contains one literal is called *unit clause*, and a literal l is called *monotone*, if $\overline{l}$ appears in no clause of ϕ.

In lines 2-7 the reduction function is applied whenever a unit clause or a monotone literal is found. The subsequent recursive call is performed in lines 11, 13 respectively. "Easy" input instances imply that the DLL algorithm requires few backtracking attempts because clauses and literals can be efficiently eliminated by means of $R(\phi, l)$ without executing many subsequent recursive calls. On the other hand, "hard" instances imply that many recursive calls or backtracking attempts are required. In the next section, the terms "easy" and "hard" are clarified. More specifically, it will be shown that parameters $|\mathcal{S}|, l$ and r specify the ratio of the number of clauses to variables of the $\widehat{\phi}_{rcb}$ instances and therefore characterize the computational complexity of the DLL algorithm.

4.1 Phase Transition in *k*-CNF Satisfiability

The k-CNF satisfiability problem is $\mathcal{NP}$-complete for $k > 2$, however, this fact does not imply that *all* instances of the k-CNF satisfiability problem are intractable to solve. In point of fact, there exists many problem instances which are "easy" to solve, i.e. one can efficiently decide whether the instance is satisfiable or is unsatisfiable. On the other hand there also exists problem instances

[5] The DLL algorithm is sometimes also called DPL or DPLL algorithm [15,16].

Algorithm 2. Davis-Logemann-Loveland algorithm (DLL($\cdot$))

```
input  : $\phi$ (Boolean formula in CNF)
output: SATISFIABLE or UNSATISFIABLE
begin
    forall unit clauses $\{l\}$ in $\phi$ do
        $\phi \leftarrow R(\phi, l)$
        if $\phi$ includes empty clause then
            return UNSATISFIABLE
    forall monotone literals $l$ in $\phi$ do
        $\phi \leftarrow R(\phi, l)$
    if $\phi$ is empty then
        return SATISFIABLE
    choose a literal $l$ in $\phi$
    if DLL($R(\phi, l)$) = SATISFIABLE) then
        return SATISFIABLE
    if DLL($R(\phi, \bar{l})$) = SATISFIABLE) then
        return SATISFIABLE
    return UNSATISFIABLE
end
```

which are "hard", i.e. one can *not* efficiently decide whether the instance is satisfiable or is not satisfiable. The computational "hardness" of finding assignments sets for randomly generated instances is characterized by the ratio [19]

$$r_k = \frac{\text{number of clauses}}{\text{number of variables}}. \tag{12}$$

If the Boolean formula ϕ has many variables and few clauses, then ϕ is *under-constrained* and as a result it exists many assignment sets. The DLL algorithm requires for under-constrained problem instances few backtracking attempts and therefore "easily" deduces the satisfiability. On the other hand, if the ratio of the number of clauses to variables is large, then ϕ is *over-constrained* and almost has no satisfying assignment set. Such over-constrained instances are likewise "easily" deducible for the DLL algorithm. However, there also exists a transition from under-constrained to the over-constrained region. In such a *phase transition* region the probability of the instances being satisfiable equals 0.5 and thus one has the largest uncertainty whether the instances are satisfiable or are unsatisfiable.

For the 3-CNF satisfiability problem, the ratio (phase transition threshold) is experimentally approximated by ≈ 4.24 [15,20]. That means, when r_3 is close[6] to 4.24, the DLL algorithm has to backtrack most frequently to determine the

[6] It is still an open problem to prove where the *exact* phase transition threshold is located. Latest theoretical work showed that the threshold r_k lies within the boundary $2.68 < r_k < 4.51$ for $k = 3$ [21].

final result. If the Boolean formula is under-constrained ($r_3 < 4.24$) or over-constrained ($r_3 > 4.24$), then the algorithm prunes whole branches from the decision tree without exploring the leaves and terminates after few recursive calls.

5 Experiment with $\widehat{\phi}_{rcb}$ Instances

The computational complexity of finding detectors is experimentally investigated with the DLL algorithm. More specifically, the parameters $l = 75, r = 3$ are chosen and $|\mathcal{S}|$ is varied from 1 to 25, i.e. for each cardinality value from 1 to 25, $\mathcal{S}$ contains distinct bit strings which are randomly drawn from $\mathcal{U}$. As a result, one obtains Boolean formulas $\widehat{\phi}_{rcb}$ in 75-3-CNF with 75 variables and $(75 - 3 + 1) \cdot |\mathcal{S}|$ clauses, $\mathbf{E}[|\widehat{\phi}_{rcb}|]$ distinct clauses, respectively. To obtain a large number of different $\widehat{\phi}_{rcb}$ instances, for each value of $|\mathcal{S}|$, 300 instances are randomly generated. The DLL algorithm is applied on each instance and the results: satisfiable/unsatisfiable and the number of backtracking attempts are noted. The result is depicted in figure 2. The abscissa denotes the ratio of the average number of distinct clauses to variables. The ordinate denotes the number of backtracking attempts (computational costs). The resulting ordinate values are colored gray if the DLL algorithm outputs satisfiable, otherwise it outputs unsatisfiable and the values are colored black.

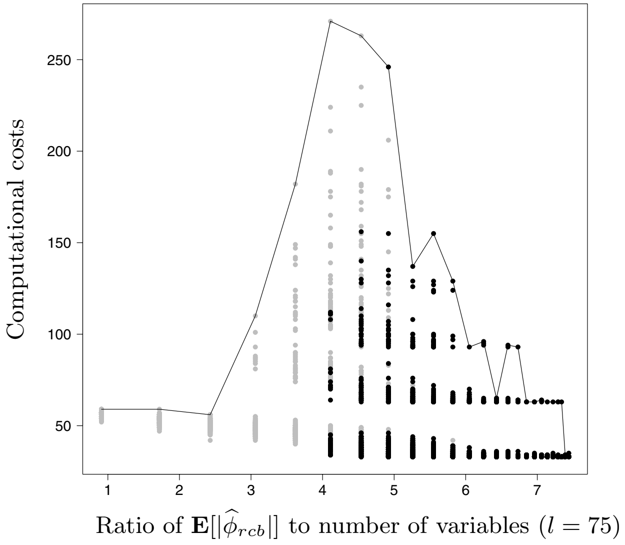


Fig. 2. Number of backtracking attempts (computational costs) of the DLL algorithm to decide whether a $\widehat{\phi}_{rcb}$ instance is satisfiable or unsatisfiable. The gray points denote satisfiable instances whereas black points denote unsatisfiable instances. The "hardest" instances are lying in the interval 4 to 5, termed phase transition region.

One can see in figure 2 that for ($r_3 < 4$) a large number of satisfiable instances exist. Or to say it the other way around, for small values of $|\mathcal{S}|$ the resulting Boolean formula $\widehat{\phi}_{rcb}$ is under-constrained and therefore a large number of satisfiable instances exist. The DLL algorithm hence "easily" deduces a satisfiability result. The number of satisfiable and unsatisfiable instances is nearly equal for ($4 < r_3 < 5$). These instances have the largest uncertainty for the DLL algorithm. As a consequence, the DLL algorithm requires the most backtracking attempts to determine whether the instances are satisfiable or are unsatisfiable. A ratio ($r_3 > 5$) implies that a large number of over-constrained instances exist and hence, the DLL algorithm "easily" deduces the unsatisfiable result. Another way to visualize this "easy-hard-easy" pattern, is to plot the percentage of satisfiable instances on the ordinate (see Fig. 3). One can see that the probability of the instances being satisfiable equals 0.5 when ($4 < r_3 < 5$) and rapidly changes to 1 for ($r_3 < 4$) and to 0 for ($r_3 > 5$).

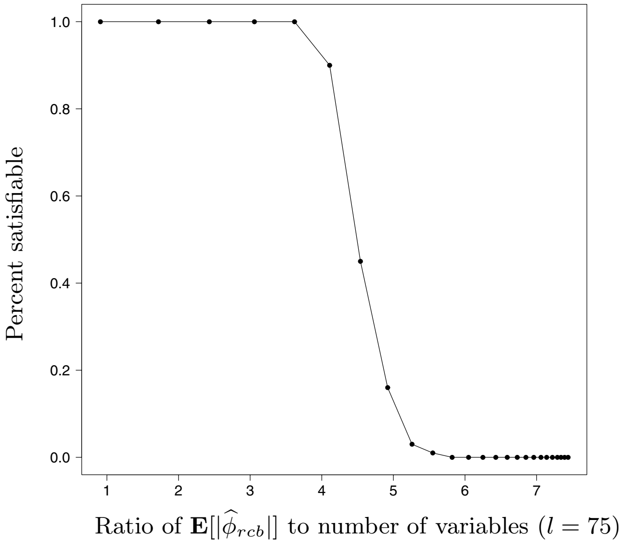


Fig. 3. Coherence between the percentage of satisfiable instances and the ratio of $\mathbf{E}[|\widehat{\phi}_{rcb}|]/l$. The "hardest" instances live in the region where the number of satisfiable and unsatisfiable instances is equal, or in other words, the probability of instances being satisfiable equals 0.5.

6 Conclusion

We have rigorously analyzed the feasibility of generating detectors with respect to the number of self bit strings $|\mathcal{S}|$, the bit string length l and matching length r. With high probability detectors either can be generated or in contrast, can not be generated. However, there also exists a region where the probability rapidly changes from 0 to 1. This behavior can be explained by transforming the problem of finding detectors into the k-CNF satisfiability problem. If a large number of self bit strings exist and r is close to 0, then the resulting Boolean formula is over-constrained and no assignment sets exist. In contrast, if a small number of self bit strings exist and r is close to l, then the resulting Boolean formula is under-constrained and as a result a large number of assignment sets exist. Moreover we exploited the problem transformation to investigate the computational complexity of finding detectors by means of the DLL algorithm. Finding detectors is “easy” for under-constrained Boolean formulas. It is also “easy” to determine for over-constrained Boolean formulas that no detectors exist. However, for parameter combinations of $|\mathcal{S}|, l$ and r where the resulting ratio of the average number of distinct clauses to variables is close to the phase transition threshold, finding detectors is “hardest”. For such “hard” instances the DLL algorithm requires the most backtracking attempts, because the probability of the instances being satisfiable equals 0.5 and thus one has the largest uncertainty whether the instances are satisfiable or are unsatisfiable.

Acknowledgment

The author thanks Erin Gardner for her valuable suggestions and comments.

References

1. Percus, J.K., Percus, O.E., Perelson, A.S.: Predicting the size of the T-cell receptor and antibody combining region from consideration of efficient self-nonself discrimination. Proceedings of National Academy of Sciences USA 90, 1691–1695 (1993)
2. D’haeseleer, P., Forrest, S., Helman, P.: An immunological approach to change detection: algorithms, analysis, and implications. In: Proceedings of the Symposium on Research in Security and Privacy, pp. 110–119. IEEE Computer Society Press, Los Alamitos (1996)
3. 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)
4. Ayara, M., Timmis, J., de Lemos, R., de Castro, L.N., Duncan, R.: Negative selection: How to generate detectors. In: Proceedings of the 1nd International Conference on Artificial Immune Systems (ICARIS), pp. 89–98. University of Kent at Canterbury Printing Unit (2002)
5. Stibor, T., Timmis, J., Eckert, C.: On the appropriateness of negative selection defined over hamming shape-space as a network intrusion detection system. In: Proceedings of Congress On Evolutionary Computation (CEC), pp. 995–1002. IEEE Press, New York (2005)
6. Stibor, T., Timmis, J., Eckert, C.: The link between r-contiguous detectors and k-CNF satisfiability. In: Proceedings of Congress On Evolutionary Computation (CEC), pp. 491–498. IEEE Press, New York (2006)
7. Feller, W.: An Introduction to Probability Theory and its Applications, 3rd edn. vol. 1. John Wiley & Sons, West Sussex, England (1968)

8. Ranang, M.T.: An artificial immune system approach to preserving security in computer networks. Master's thesis, Norges Teknisk-Naturvitenskapelige Universitet (2002)
9. Uspensky, J.V.: Introduction to Mathematical Probability. McGraw-Hill, New York (1937)
10. Freitas, A.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)
11. 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)
12. Stibor, T., Timmis, J., Eckert, C.: Generalization regions in hamming negative selection. In: Intelligent Information Processing and Web Mining. Advances in Soft Computing, pp. 447–456. Springer, Heidelberg (2006)
13. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. MIT Press, Cambridge, MA (2002)
14. Kullmann, O.: The SAT, solver competition on random instances. Journal on Satisfiability, Boolean Modeling and Computation 2, 61–102 (2006)
15. Freeman, J.W.: Hard random 3-SAT problems and the Davis-Putnam procedure. Artificial Intelligence 81(1-2), 183–198 (1996)
16. Ouyang, M.: How good are branching rules in DPLL. Discrete Applied Mathematics 89(1-3), 281–286 (1998)
17. Davis, M., Logemann, G., Loveland, D.: A machine program for theorem-proving. Communications of the ACM 5(7), 394–397 (1962)
18. Davis, M., Putnam, H.: A computing procedure for quantification theory. Journal of the ACM (JACM) 7(3), 201–215 (1960)
19. Gent, I.P., Walsh, T.: The SAT phase transition. In: Proceedings of the 11th European Conference on Artificial Intelligence, pp. 105–109. John Wiley & Sons, West Sussex, England (1994)
20. Selman, B., Mitchell, D.G., Levesque, H.J.: Generating hard satisfiability problems. Artificial Intelligence 81(1-2), 17–29 (1996)
21. Achlioptas, D., Naor, A., Peres, Y.: Rigorous location of phase transitions in hard optimization problems. Nature 435, 759–764 (2005)

Real-Valued Negative Selection Algorithm with a Quasi-Monte Carlo Genetic Detector Generation

Jorge L.M. Amaral[1], José F.M. Amaral[1], and Ricardo Tanscheit[2]

[1] Dept. of Electronics and Telecommunications Engineering,
Rio de Janeiro State University, 20550-013 Rio de Janeiro, RJ, Brazil
{jamaral,franco}@uerj.br
[2] Dept. of Electrical Engineering, Catholic University of Rio de Janeiro,
22.453-900 Rio de Janeiro, RJ, Brazil
ricardo@ele.puc-rio.br

Abstract. A new scheme for detector generation for the Real-Valued Negative Selection Algorithm (RNSA) is presented. The proposed method makes use of genetic algorithms and Quasi-Monte Carlo Integration to automatically generate a small number of very efficient detectors. Results have demonstrated that a fault detection system with detectors generated by the proposed scheme is able to detect faults in analog circuits and in a ball bearing dataset.

Keywords: Negative Selection, Detector Generation, Genetic algorithms, Quasi-Monte Carlo.

1 Introduction

The development of test strategies for detecting and diagnosing faults is still severely dependent on engineers' expertise and on the knowledge they have about the system's operational characteristics. As a result, fault detection and identification is still an interactive and time-consuming process. A survey of research in the area shows that, in the last decades, a good amount of research on fault diagnosis has concentrated on the development of intelligent tools to make the task of fault diagnosing easier. Although there have been several important developments, these new technologies have not been widely accepted [1]. This should motivate the researchers to look for other paradigms and to develop new strategies for fault diagnosis.

Artificial immune systems [2] take their inspiration from the operation of the human immune system, which is capable of recognizing virtually any pathogenic agent. This is done by distinguishing the body's own cells and molecules (*self*) from foreign antigens (*nonself*). Despite the fact that this is a relatively new topic of research, applications in many areas already exist [2], e.g., computer security, virus protection, anomaly detection, process monitoring, pattern recognition, robot control and software fault tolerance. The behaviors (normal operation or different types of faults) of each system can be associated with a unique set of features, i.e., a signature. The method proposed assumes that similar system' behaviors present similar signatures. The immune algorithm links the signatures that correspond to the normal

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 156–167, 2007.

behavior to the concept of *self*, so that this can be used to identify abnormal behavior. Therefore the proposed immune-based fault detection system is capable of determining if the system under test is faulty. An important feature of the proposed method is that it can automatically generate very efficient detectors by genetic algorithms.

The material in this paper is arranged in the following order. In section 2 artificial immune systems are briefly reviewed, with emphasis on the Negative Selection Algorithm (NSA). In section 3 Genetic Algorithms are briefly reviewed. In section 4 the proposed method for generating a small and efficient number of detectors for the NSA is described. The results obtained for fault detection in a Sallen-Key bandpass filter, in a continuous-time state variable filter and in ball bearing dataset are discussed in section 5. Finally, section 6 concludes this work.

2 Negative Selection Algorithm

Artificial immune systems take their inspiration from the operation of the human immune system to create novel solutions to problem solving. Some algorithms have been presented as successful implementations of artificial immune systems: the immune network, the Clonal Selection Algorithm and the Negative Selection Algorithm [2]. The Negative Selection algorithm can be applied to several areas and is especially useful in fault detection. This algorithm makes use of the immune system's property of distinguishing any foreign cells (*nonself*) from the body’s own cells [3]. This characteristic can be used for distinguishing normal systems patterns from abnormal ones, thus providing a fault detection mechanism. It is important to identify situations where this characteristic could be an advantage [4]: when the normal behavior of a system is defined by a set of complex patterns, where it is very difficult to obtain their relations – it may be easier to look at the abnormal patterns instead of the normal ones – and when the number of possible abnormal patterns is much larger than that of normal ones. Since training a fault detection system with a large number of fault situations becomes unpractical, it is a good strategy to first detect any abnormal behavior and then try to identify its cause.

The negative selection algorithm – NSA [3] is inspired by the mechanism used by the immune system to train the T-cells to recognize antigens (*nonself*) and to prevent them from recognizing the body’s own cells (*self*). It generates a set of (binary) detectors by randomly creating candidates and then discarding those that correspond to the notion of self. These detectors can later be used for detecting anomalies. In the original proposal, the NSA algorithm has three basic steps: *Self* Definition – a collection of binary strings S of limited length that needs protection or monitoring; Detector Generation – set of detectors that fail to match any strings in S; Monitoring – monitor S for changes by continually matching the detectors against S; if any detector ever matches it, then an anomaly must have occurred, since the detectors are designed to never match any of the strings in S. The drawbacks found in the NSA with binary representation have encouraged the development of a Real Valued Negative Selection Algorithm (RNSA) [5], which has the same three basic steps of the original NSA but employs real-valued encoding for the characterization of *self* and *nonself*. The detectors and *self* points can be seen as hyperspheres; the *self* points represent normal behavior and the detectors are responsible for finding anomalies. The detector's radius determines the threshold for the matching rule, which in this case is the Euclidean distance. This algorithm takes the *self* samples, which are represented by n-dimensional points, and tries to evolve detectors (another set of n dimensional points) to cover the *nonself* space. This is performed through an interactive process aiming at two main goals: keep the detectors away from the *self* set and maximize the nonself space coverage by keeping the detectors apart. The hyperspace covered by each detector can be set through a parameter *r* (radius). Once detector generation is completed, the detectors can be employed in the third phase (Monitoring). Figure 1 shows a two dimensional example of the Detector Set and of the self points. The gray area represents the *self* region. The circles identified by "+" represent the detectors. Assuming that the point indicated by "o" represents the system's current state, no anomaly is detected, since it is not covered by the detector. However, the point indicated by "*" is reported as an anomaly (fail) since it is located in the space covered by the detectors.

The RNSA [5] can also be used to generate detectors that work as *nonself* samples, which, together with the *self* samples, are taken as inputs to a classification algorithm.

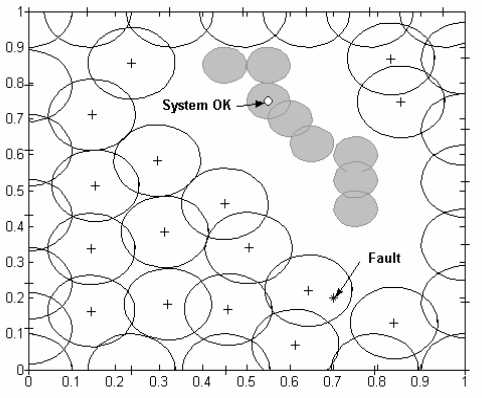


Fig. 1. Detector Coverage

In the original RNSA, the number of detectors to be used has to be chosen beforehand, there is no way to determine if the algorithm is really improving the placement of the detectors in order to provide the best possible distribution. In an improved version of the algorithm [6], the number of detectors can be estimated beforehand and their positions are determined by the minimization of a function that represents the overlap between the detectors themselves and between them and the points that represent the *self*. In these implementations of RNSA, all detectors have the same radius, which may pose a scalability limitation: if a small radius is chosen, a large number of detectors will be needed to cover the space; if a large radius is chosen, parts of the *nonself* space may be left uncovered. This limitation encouraged the development of new detector representations – allowing variable geometry – and of new detector generation schemes. As examples of such schemes, the V-Detectors [7], the boundary-aware negative selection algorithm [8] and the Quadtree automatic detector generation [9] can be mentioned.

3 Genetic Algorithms

Genetic Algorithms [10] provide an adaptive searching mechanism inspired on Darwin's principle of reproduction and survival of the fittest. The individuals (solutions) in a population are represented by chromosomes; each of them is associated to a fitness value (problem evaluation). The chromosomes are subjected to an evolutionary process that takes several cycles (generations). Basic operations are selection, reproduction, crossover and mutation. Parent selection gives a higher probability of reproduction to the fittest individuals. During crossover some reproduced individuals cross and exchange their genetic characteristics. Mutations may occur in a small percentage and cause a random variation in the genetic material, thus contributing to introduce variety in the population. The evolution process guides the genetic algorithm through more promising regions in the search space. Some of the advantages of using genetic algorithms are: it is a global search technique, can be applied to the optimization of ill-structured problems and do not require a precise mathematical formulation for the problem. Genetic algorithms are robust, applicable to a number of problems and efficient, in the sense that either a sub-optimal or optimal solution may be found within reasonable time and computational effort.

4 Proposed Method

The proposed method makes use of genetic algorithms to generate the detector set. The success of this method depends on how the solution is coded in the chromosome and on the fitness function chosen for evaluation of the solution. The next two subsections explain each of these implementations.

4.1 Chromosome

In the proposed method, each chromosome represents a possible detector set where each gene represents a pointer (index) to a certain point in a sequence of samples that represents a probability distribution. Therefore, each chromosome determines which points from the sequence should be used as detectors' centers. The main idea behind this representation is to reduce the chromosome size, especially in high dimension problems. Irrespective of the problem's dimension, the chromosome size is equal to the maximum number of detectors desired. A possible example of this representation is shown in Fig. 2, where the circles represent the *self* set; the points identified by '+' are candidates to be detectors' centers. In the chromosome shown on the right the five genes indicate that points 3, 6, 10, 4 and 8 are the candidates.

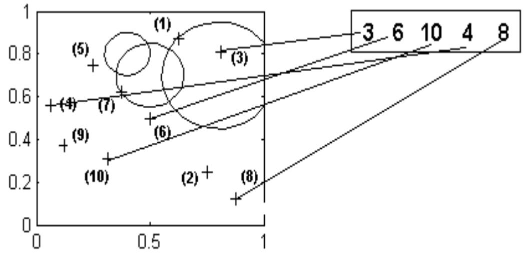


Fig. 2. Circles represent the *self* set; points identified by "+" represent possible detectors' centers; the chromosome (on the right) indicates the points that will be used as detectors' centers

In this work, detectors' centers were placed by using quasi-random sequences, since these show a high level of uniformity in multidimensional spaces. Some of well known quasi-random sequences are Faure, Sobol and Halton [11]. An example of the advantage of using quasi-random distributions can be seen in Fig. 3, which shows the volume (area) covered by a detector set in a hypercube as a function of the number of detectors for three different distributions. All detectors have the same radius (0.05), and the detectors' centers are taken the following distributions: pseudo-random (1), Faure quasi-random (2) and Sobol quasi-random (3). The volume (area) covered when detectors' centers are taken from quasi-random distributions (2 and 3), is larger than the volume (area) covered when detectors' centers are taken from a pseudo-random distribution (1).

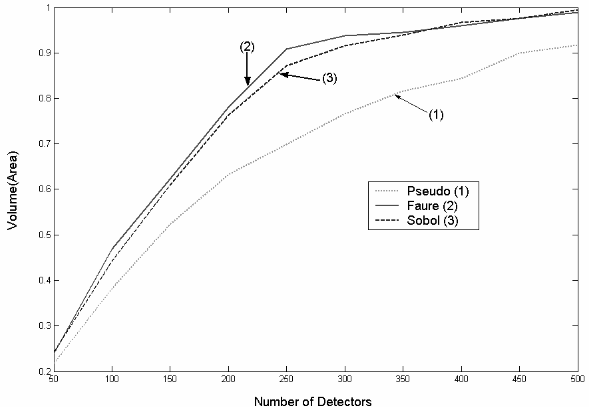


Fig. 3. Volume covered by a detector set versus no. of detectors

Once the detectors' centers have been determined, a decoding function computes the largest possible radius, taking into consideration that the Detector Set cannot "attack" the *self* set and that there should be a certain amount of overlap between the detectors. If one of the chosen centers is inside the *self* set, it is discarded. For example, in Fig. 2, the detector center with label (3) would be discarded. After the decoding function has calculated the largest possible radius that does not “attack” the *self* set, the radius could be adjusted to have a certain degree of overlap. The parameter α controls the amount of overlap. If α = 0, the calculated radius is equal to the distance between the detectors; if α = 1 there is no overlap between the detectors. The calculated radius for a detector based on the value of α is shown in Fig. 4.

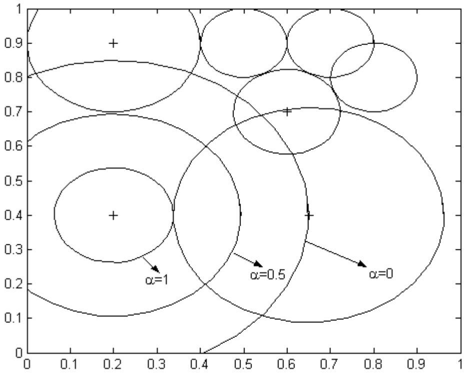


Fig. 4. Detector radius as a function of the parameter α: circles identified by "+" represent the detectors; remaining circles represent the *self* set

The process of detectors placement after the radius has been calculated by the decoding function is shown in Fig. 5 (from the upper left, clockwise). Initially, the decoding function establishes the radius of the first detector so that it will not overlap with any of the circles that represent the *self* set. A second detector is placed with a certain amount of overlap with the first one. This rule – a certain overlap with other detectors and no overlap at all with the self set – is observed in further placements.

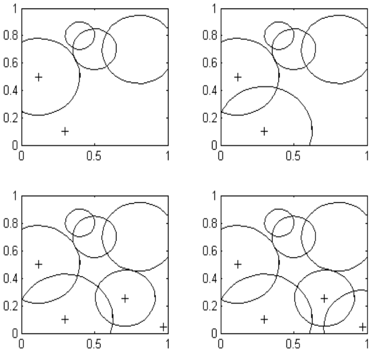


Fig. 5. Detector Placement: circles identified by "+" represent the detectors; remaining circles represent the *self* set

4.2 Fitness Function Computation by Monte Carlo Integration

The fitness function used to evaluate the quality of the solution coded in the chromosome is the volume of the detector set. If the volume of the *self* set V_s can be computed, the volume V_{ns} that should be covered by the detector set can easily be calculated as $V_{ns} = 1 - V_s$, since the *self/nonself* space corresponds to a unit hypercube $[0,1]^n$. Therefore, the closer the volume of the detector set represented in the chromosome is to the value of V_{ns}, the better the solution. The volume of the detector set and the volume of the *self* set can be calculated by means of a Monte Carlo Integration. This technique is used for evaluating the integral I of a function $f(x)$ [11][12], in the n-dimensional unit cube.

Here, since the volume is to be calculated, $f(X_i)$ is 0 if the point X_i is outside the volume, and 1 if X_i is inside the volume.

In its basic form [12], the Monte Carlo Integration samples N independent random points X_1, X_2, ..., X_N with uniform distribution in the unit cube to compute an estimation I' of the integral I, i.e. $E(I') = I$, as:

$$I' = \sum_{i=1}^{N} f(X_i) \tag{1}$$

Assuming that the variance of I' is σ^2, the mean square error can be written as:

$$E((I'-I)^2) = Var(I') = \frac{\sigma^2}{N} \tag{2}$$

It can therefore be concluded that the estimated integral has a standard error of $\sigma / N^{1/2}$. This means that the estimated error of the integral will decrease at the rate of $N^{1/2}$ (convergence rate). In order to get a better convergence, quasi-random distributions are employed again. As opposed to pseudo-random ones, in quasi-random distributions samples are deterministic and calculated in a way that “spaces” left between previous samples are filled by the new ones. This reduces the standard deviation and speeds up a Monte Carlo simulation [11] [12].

5 Case Studies

In order to evaluate the ability of the detector set generated by the proposed method in detecting faults, three experiments were made. The first and second case studies are related to fault detection in analog circuits and the third is related to fault detection in ball bearings.

To assess the efficiency of the detectors, a test set with normal and abnormal behavior was created for each experiment. The test set was presented to the detector set as proposed in the negative selection algorithm [3] (Monitor). The test set was also presented to a set of classifiers that were trained and validated by using points that represent the normal behavior of the case studies (*self*) set and points taken from a quasi-random distribution that lay in the detector set range, in the fashion proposed in [5]. The idea behind using these points is not to bias the classifiers towards specific types of faults, so that normal behavior can be distinguished from any faulty one. The classifiers used were: K-Nearest Neighbors (KNN), Probabilistic Neural Networks (PNN), Multilayer Perceptrons trained by the Levemberg-Marquardt method (MLP_LM) and neural networks with Radial Basis Functions (RBF).

5.1 Fault Detection in Analog Circuits

The analog circuits used in these experiments are: the Sallen Key bandpass circuit and the universal filter (Fig. 6) – a continuous-time state variable filter that incorporates high pass, band pass and low pass filters whose outputs are denoted by HPO, BPO and LPO respectively. Both circuits have been used in several publications [13],[14],[15],[16],[17] and the universal filter is considered a benchmark in fault detection, diagnosis and testing of analog circuits [16].

In this work, the circuits' step responses are applied to a differentiation circuit, which generates their impulse responses [17]. The presence of a fault or variations in circuit or device parameters may cause the impulse response (IR) to be different from that of a "golden circuit" in terms of amplitude or shape characteristics. A technique based on cross-correlation operations is employed to quantify that difference. The location of the cross-correlation peak (amplitude and time) of the two waveforms expresses the similarity between the IR of the circuit under test and of the "golden circuit". In order to evaluate the efficiency of the detectors, a test set with normal and abnormal circuits was created for each experiment.

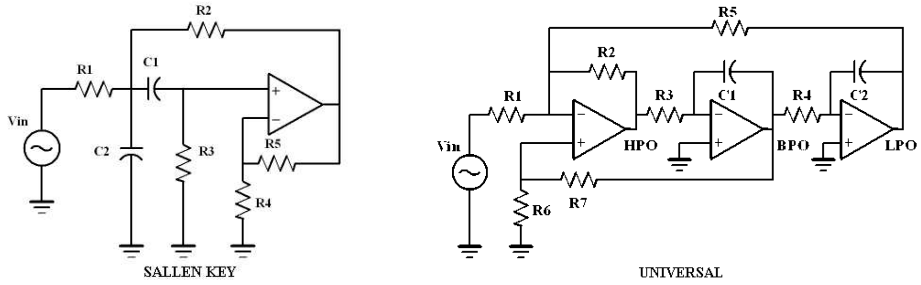


Fig. 6. The Sallen Key Bandpass circuit and the Universal Filter

The *self* sets were created by calculating the locations of the correlation peak (amplitude and time) for circuits exhibiting normal behavior, i.e., those where all components meet tolerance specifications. It was considered that radius of each element of the *self* set was the distance to its nearest neighbor. This relates the employed radius to the *self* set distribution.

The test set used for evaluating the generated detectors consisted of several normal and faulty circuits. In these, for the Sallen-Key circuit, a deviation of 50% of the components' nominal value was considered as a fault, as in [13], [14], [15]. The faults for the universal filter were catastrophic (short circuit or open circuit), as in [17].

The detector generation was performed by the proposed method (see Section 4). The genetic algorithm has been configured as: population = 100; generations = 50;

crossover = 65% and mutation = 0.8%. Table 1 shows the nonself coverage and the number of detectors for the Sallen-Key filter and for the universal filter. It is worth noting that the number of detectors required to provide near full coverage of the nonself is much smaller than the one that would be needed by the original RNSA. The estimation of the number of detectors for RNSA is done by assuming that all detectors have the same radius and that it is equal to the maximum distance between a *self* point and the nearest neighbor. This relates the employed radius for the detectors to the *self* set distribution.

Table 1. Nonself coverage and number of detectors

Circuit	*Nonself* coverage	Number of detectors	Number of detectors estimated for RNSA
Sallen Key	99.91 %	27	5602
Universal	99.97 %	18	19,521,484

Results for each of the classifiers are shown in Table 2. Parameters used in the test of the Sallen Key were: KNN – no. of neighbors = 1; PNN – spread = 0.0314192; MLP-LM – number of neurons = 4; RBF – number of units = 20. All classifiers presented a very good performance, especially the Monitor, which agrees with the excellent coverage achieved by the detectors. The results shown here are better then those found in [13], which reports an error of 11%, and agrees to those reported in [14], [15] (0%). It is worth mentioned that those classifiers were based only on the knowledge of the normal behavior. Regarding the universal filter the parameters were: KNN – no. of neighbors = 1; PNN – spread = 0.03148; MLP-LM – number of neurons = 3; RBF – number of units = 40. The very good performance of all classifiers agrees with the excellent coverage achieved by the detectors. The results shown here match those found in [17]. It is worth mentioning that the classifiers were based solely on the knowledge of the normal behavior.

Table 2. Test results for the Sallen-key filter and for the Universal Filter

Circuit	Sallen Key			Universal		
Classifier	Training Error	Validation Error	Test Error	Training Error	Validation Error	Test Error
Monitor	0	0	0	0	0	0
KNN	0	0.39%	0	0	0.15%	0.11%
PNN	0	0.79%	0	0	0	0
MLP_LM	0	0	0	0	0	0.3%
RBF	0	0	0	0	0	0

5.2 Ball Bearing Data

Fault detection in ball bearings has been the subject of works on negative selection [7] [8]. Experimental data are the times series of measured acceleration of ball bearings [18]. The two preprocessing techniques are described in [7]. The first is basically DFT, and the second, which was used to produce the results, uses statistical moments to represent the property of each segment of 128 points. The moments of the first (mean), second (variance), third, four and fifth order are used.

Detector generation was performed by the proposed method (see Section 4). The genetic algorithm has been configured as follows: population = 100; generations = 100; crossover =65% and mutation =0.8%. Table 3 shows the nonself coverage and the number of detectors. Once again, the number of detectors is much smaller than that estimated for the RNSA and the achieved coverage is near 100%.

Table 3. Nonself coverage and number of detectors – Sallen-Key

Nonself coverage	Number of detectors	Number of detectors estimated for RNSA
99.99 %	100	14,824,841

The test set for evaluation of the generated detector consists of data representing normal bearings (*self*) and outer race completely broken (*nonself*). Table 4 shows the results. Parameters used in the classifiers were: KNN – no. of neighbors = 1; PNN – spread =0.019435; MLP-LM – number of neurons = 6; RBF – number of units = 20.

Table 4. Test results for the ball bearing

Condition	Point Wise	Boundary-Aware	Monitor	KNN	PNN	MLP-LM	RBF
New	0	0.15%	0	0	0	0	0
Broken race	74.82%	93.4%	93.4%	84.69%	83.59%	86.77%	90.75%

The results achieved by the Monitor match the boundary-aware algorithm [8] and are better than those obtained by the point wise algorithm [7], which agrees with the excellent coverage achieved by the detectors. Although the trained classifiers did not perform as well as the Monitor in this case, they perform better than the point wise algorithm; the RBF classifier in particular has shown a competitive detection rate.

5 Conclusions

A new detector generation scheme has been proposed which employs genetic algorithms and Quasi-Monte Carlo Integration. One of the important features of this generation scheme is that is suited to high dimension problems. The size of the chromosome is related only to the maximum number of detectors and not to the problem dimension. The error in a Quasi-Monte Carlo integration, which computes the *nonself* coverage, is related to the number of points used and not to the problem dimension. This addresses the issue of scalability and allows the use of the Negative Selection Algorithm in more complex situations.

It is possible to implement two different fault detection systems with the detectors generated by the new scheme: the detectors may be used either as proposed in the Negative Selection Algorithm or for generating *nonself* samples; classifiers can then be trained to perform fault detection. This is especially useful when there is scarce or no information about the possible abnormal behavior of the system. One important feature is that the fault detection systems are based only on the knowledge of the system's normal behavior; there is no assumption about the kind of fault that is expected.

Currently, other feature extraction methods are under investigation, with the aim of developing more general preprocessing steps. Besides, there are some aspects of this detector generation scheme that need further investigation such as: other Quasi-Monte Carlo algorithms, the influence of the α parameter and generation of the *nonself* samples to train classifiers.

Acknowledgments. The authors wish to thank Dr. Zhou Ji and Dr. Dipankar Dasgupta for making the ball bearing dataset available.

References

1. Fenton, W.G., McGinnity, T.M., Maguire, L.P.: Fault diagnosis of electronic systems using intelligent techniques: a review. IEEE Transactions on Systems, Man and Cybernetics – Part C 31, 269–281 (2001)
2. Castro, L.N., Timmis, J.: Artificial Immune System: A New Computational Intelligence Approach. Springer, London (2002)
3. Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R.: Self-nonself discrimination in a computer. In: Proc. IEEE Symposium on Research in Security and Privacy, Oakland, pp. 202–212 (1994)
4. Martins, J.F., Costa Branco, P.J., Dente, J.A.: Fault detection using immune-based systems and formal language algorithms. In: Proc. 39th Conference on Decision and Control, Sidney, Australia, pp. 2633–2638 (2000)
5. Gonzalez, F., Dasgupta, D., Kozma, R.: Combining negative selection and classification techniques for anomaly detection. In: Proc. Congress on Evolutionary Computation, Hawaii, pp. 705–710 (2002)
6. Gonzalez, F., Dasgupta, D., Nino, L.F: A randomized real-valued negative selection algorithm. In: Proc. 2nd International Conference on Artificial Immune Systems, Edinburgh, UK, pp. 261–272 (2003)
7. Ji, Z., Dasgupta, D.: Real-valued negative selection algorithm with variable size detectors. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 287–298. Springer, Heidelberg (2004)
8. Ji, Z.: A boudary-aware negative selection algorithm. In: Proceedings of IASTED International Conference on Artificial Intelligence and Soft Computing (ASC 2005) (2005)
9. Amaral, J.L.M., Amaral, J.F.M., Tanscheit, R.: An Immune Fault Detection System for Analog Circuits with Automatic Detector Generation. IEEE Congress on Evolutionary Computation, 2966–2972 (2006)
10. Goldberg, D.E: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, London, UK (1989)
11. Morokoff, W.J., Caflisch, R.E.: Quasi-Monte Carlo Integration. Academic Press, San Diego (1993)

12. Levy, G.: Where Numerics Matter: An introduction to quasi-random numbers. Financial Engineering News, Issue 24, World Wide Web (2002), http:// www.fenews.com/ fen24/ levy.html
13. Amaral, J.L., Amaral, J.F., Tanscheit, R., Pacheco, M.A.C: An immune inspired fault diagnosis system for analog circuits using wavelet signatures. In: Proc. NASA/DoD Conference on Evolvable Hardware, Seattle, Washington, pp. 138–141 (2004)
14. Spina, R., Upadhyaya, S.: Linear circuit fault diagnosis using neuromorphic analyzers, IEEE Transactions on Circuits and Systems II, 44, 188–196, March 199719. Galanti, S., Jung, A.: Low-Discrepancy Sequences: Monte Carlo Simulation of Option Prices. Journal of Derivatives, 63–83 (1997)
15. Aminian, M., Aminian, F.: Neural-network based analog-circuit fault-diagnosis using wavelet transform as preprocessor. IEEE Transactions on Systems, Man and Cybernetics 47(2), 151–156 (2000)
16. Kaminska, B., Arabi, K., Bell, I., Goteli, P., Huertas, J.L., Kim, B., Rueda, A., Soma, M.: Analog and mixed-signal benchmark circuits – first release. In: Proc. IEEE International Test Conference, Washington, DC, pp. 183–1900. IEEE Computer Society Press, Los Alamitos (1998)
17. Singh, A., Patel, C., Plusquellic, J.: On-chip impulse response generation for analog and mixed-signal testing. In: Proc. IEEE International Test Conference, Charlotte, NC, pp. 262–270. IEEE Computer Society Press, Los Alamitos (2004)
18. Structural Integrity and damage assessment network. World Wide Web (2004), http:// www.brunel.ac.uk/research/cnca/sida/html/data.html

A Novel Fast Negative Selection Algorithm Enhanced by State Graphs

Wenjian Luo[1,2], Xin Wang[1], and Xufa Wang[1,2]

[1] Nature Inspired Computation and Applications Laboratory, Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China
[2] Anhui Key Laboratory of Software in Computing and Communication, University of Science and Technology of China, Hefei 230027, China
wjluo@ustc.edu.cn, sinbarwx@mail.ustc.edu.cn, xfwang@ustc.edu.cn

Abstract. Negative Selection Algorithm is widely applied in Artificial Immune Systems, but it is not fast enough when there are mass data need to be processed. Multi-pattern matching algorithms are able to locate all occurrences of multi-patterns in an input string by just one scan operation. Inspired by the multi-pattern matching algorithm proposed by Aho and Corasick in 1975 [1], a novel fast negative selection algorithm is proposed for the "r-contiguous-bits" matching rule in this paper. The algorithm constructs a self state graph and a detector state graph according to the self set and the detector set respectively, and processes input strings using partial matching algorithm based on the state graph. The time complexity of this algorithm when processing an input string of length l is O(l). Experiments are carried out to make comparisons on the time and space costs between this new algorithm and the traditional negative selection algorithm.

1 Introduction

Negative Selection Algorithm (NSA) is a change-detection algorithm, inspired by the negative selection mechanism in the course of T-Cells maturation in biological immune system [2]. It has been widely applied in many Artificial Immune Systems recently.

Inspired by the multi-pattern matching algorithm proposed by Aho and Corasick in 1975 [1], a novel fast negative selection algorithm is proposed in this paper. This new algorithm is presented for the "r-contiguous-bits" partial matching rule, i.e. two strings match each other if and only if they have identical symbols in at least r contiguous positions. For convenience, this novel algorithm is named as G-NSA as it is based on a graph data structure. Comparisons between G-NSA and traditional NSA are conducted according to the results of the experiments carried out in this paper. The experimental results demonstrated that, compared to traditional NSA, G-NSA has much lower time complexity of processing an input string.

The rest of this paper is organized as follows. A brief review of multi-pattern matching algorithms is given in section 2. The G-NSA is introduced in section 3. Section 4 includes the experiments and the results. Section 5 briefly discusses related works and G-NSA. A brief conclusion of this paper is given in section 6.

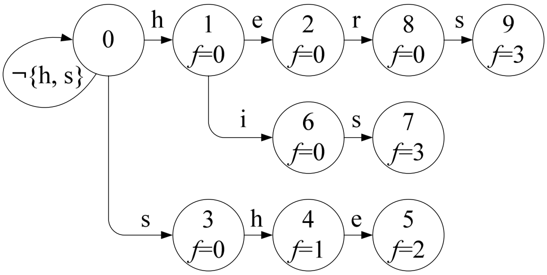
L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 168–181, 2007.

2 Multi-pattern Matching Algorithms

Multi-pattern matching algorithms are able to locate all occurrences of a finite number of patterns in an input string by just one scan operation. The first multi-pattern matching algorithm based on a finite state string pattern matching machine was proposed by Aho and Corasick in 1975 [1]. The algorithm consists of two phases: (1) Constructing a finite state pattern matching machine from the keywords set. (2) Applying the pattern matching machine to recognize the occurrences of the keywords in an input text string.

The pattern matching machine is the most important data structure in the multi-pattern matching algorithm. The pattern matching machine in [1] is composed of three kinds of functions as follows: (1) The goto function, denoted by g. (2) The failure function, denoted by f. (3) The output function, denoted by *output*.

Because of the page limits, the details of these functions are omitted in this paper. In Fig. 1, these functions in the pattern matching machine constructed for the set of keywords {he, she, his, hers} are illustrated [1].



(a) Goto and failure functions

i	2	5	7	9
output (i)	{he}	{she, he}	{his}	{hers}

(b) Output functions

Fig. 1. The pattern matching machine constructed for the set of keywords {he, she, his, hers}. This example is extracted from [1]. The state nodes of this machine are numbered as from 0 to 9. The state 0 is designated as the start state. The goto and failure functions are illustrated in Fig. 1 (a), and the output functions are given in Fig. 1 (b).

Aho and Corasick have proved that the goto, failure and output functions constructed from a given set of keywords K are valid for K. The time complexities of constructing these functions of the pattern matching machine are proportional to the sum of the lengths of the keywords. And the number of state transitions made by the pattern matching machine is fewer than $2n$ when processing an input string with length of n.

Moreover, in [1], Aho and Corasick improved the pattern matching machine by eliminating all failure transitions, and using the *next move* function δ of a deterministic finite automaton instead of the *goto* and *failure* functions. A deterministic finite automaton makes just one state transition for each symbol of the input string. In this way, the pattern matching machine makes no more than *n* state transitions in processing an input string of length *n* [1].

Based on the algorithm proposed by Aho and Corasick, some improved algorithms have been introduced. Wu and Manber proposed an efficient and flexible algorithm for approximate string matching in [3] to support searching with errors, and a fast multi-pattern matching algorithm in [4]. Wang and his colleagues presented faster multi-pattern matching algorithms that can skip as many characters as possible by making full use of the information in the matching failure when process an input string [5]. Tuck and his colleagues proposed a modified algorithm that drastically reduced the amount of memory required and improved its performance on hardware implementations [6].

3 A Novel Fast NSA Based on State Graphs

The efficient multi-pattern matching algorithm proposed by Aho and Corasick [1] is adopted to develop a novel negative selection algorithm, namely G-NSA, for the "r-contiguous-bits" matching rule in this paper. In general, G-NSA has the followings phases:

(1) The self set is converted to a state graph, namely, the self graph.
(2) Based on the self graph, the detectors are generated efficiently. Here two different approaches for generating candidate detectors can be adopted. One is to generate the candidate detectors randomly, and the other is to generate the candidate detectors with a heuristic algorithm (given as **Algorithm C** in the later part of this section). If the candidate detector is matched by the self graph, it will be deleted. Otherwise, it will be added into the detector set.
(3) The detector set is converted to a state graph, namely, the detector graph.
(4) For each string to be detected, the partial matching operation is performed base on the detector graph. If matched, the string is asserted to be an abnormal one.

Some symbols frequently used in the following sections are defined here:

$\boldsymbol{A}$: the coding alphabet.
$\boldsymbol{m}$: size of A, namely the number of coding symbols.
$\boldsymbol{l}$: coding length of the self strings and detector strings.
$\boldsymbol{r}$: partial matching length threshold, namely the least number of contiguous matching bits required for a match.
$\boldsymbol{S}$: the set of self strings.
$\boldsymbol{N_S}$: the number of self strings in S.
$\boldsymbol{R}$: the set of detectors.
$\boldsymbol{N_R}$: the number of detectors in R.
$\boldsymbol{d}$: a detector string.

The Algorithm 2, Algorithm 3 and Algorithm 4 in [1] are adopted to construct a state graph from a set of strings, i.e. the self set or detector set. However, the negative selection algorithm is different from traditional multi-pattern matching algorithms. The main differences are given as follows.

(1) Generally, in NSA, when an anomaly string is matched by the detector set, it is not necessary to indicate which detector has matched the string. Consequently, the *output* function in [1] is not needed by G-NSA. In this paper, the *output* function is not considered. Anyway, if the detector is need, the *output* functions still can be reserved.

(2) When "r-contiguous-bits" matching rule is adopted in NSA, the positions of the r-bit patterns in a detector should be recorded in the graph state nodes. Therefore, the Algorithm 2 in [1] should be modified to satisfy this change. That is to say, the index of

```
begin
    $newstate := 0$
    for (every input string $x$) do  $enter(x)$
    for (all $a$ with $g(0, a)=fail$) do  $g(0, a) := 0$
    perform Algorithm 3 in [1] to construct the $failure$ function $f$
    perform Algorithm 4 in [1] to construct the $next\ move$ function $\delta$
end

procedure $enter(x)$, $x=\{a_1, a_2, \cdots , a_l\}$:
begin
    $i := 1$
    while $i <= l-r+1$ do
        begin
            $state := 0$
            $j := i$
            while $g(state, a_j) \neq fail$ and $j < i+r$ do
                begin
                    $state := g(state, a_j)$
                    $j := j+1$
                end
            while $j < i+r$ do
                begin
                    $newstate := newstate + 1$
                    $g(state, a_j) := newstate$
                    $state := newstate$
                    $layer(state) := j-i+1$
                    $j := j+1$
                end
            if $j \notin endposition(state)$    then    $endposition(state) \cup \{j\}$
            $i := i+1$
        end
end
```

Fig. 2. Algorithm A: Constructing a state graph from a set of strings. This algorithm is designed by improving the Algorithm 2 in [1] according the requirements of G-NSA. The Algorithm 3 and Algorithm 4 in [1] are also adopted here.

the last position of every r-bit pattern in a detector should be recorded in an *end position* array in the corresponding state node of a graph.

(3) Moreover, to simplify the process of the heuristic detector generating algorithm introduced in this paper, a *layer* function is added to every state node of a graph, which is used to indicate the shortest distance from the current state node to the start state.

Therefore, the pattern matching machine for G-NSA is composed of three kinds of functions as follows: the *next move* function, the *endposition* function, and the *layer* function.

The **Algorithm A** given in Fig. 2 is used for constructing a state graph from a set of strings (the self set or the detector set). In **Algorithm A**, the *next move* function and the *layer* function are constructed for each node, and the *endposition* function is constructed only for each node on layer *r*.

In Fig.3, the partial matching algorithm based on a state graph is described, i.e. the **Algorithm B**. The **Algorithm B** is not only used for filtering the randomly generated candidate detectors in the phase of generating the detector set, but also used for monitoring the anomaly strings.

```
begin
    $state$ := 0
    for $i$ := 1 until $l$ do
        begin
            $state := \delta(state, a_i)$
            if $layer(state) = r$    then
                if $state \in endposition(state)$    then    return $Matched$
        end
    return $Unmatched$
end
```

Fig. 3. Algorithm B: The "r-contiguous-bits" partial matching algorithm based on a graph. An input string x ($a_1, a_2, \cdots, a_l$) is traced in the graph to search for a match.

Based on a self graph, a simple approach to the detector set generating can be given as follows: (1) Generating some random strings as candidate detectors. (2) Performing the **Algorithm B** between the candidate detectors and the self graph, and deleting the candidate detectors that get the *Matched* result. And then the rest of the candidate detectors are mature detectors.

Based on the self graph, a heuristic algorithm also can be developed to generate the detector set, as shown in Fig. 4, namely **Algorithm C**. In this heuristic algorithm, the symbols of a new detector are generated one by one, and every symbol is randomly selected from the coding alphabet or a subset of the coding alphabet that does not result in matching with the self patterns stored in the self graph.

```
begin
    state := 0
    i := 1
    while i <= l do
        begin
            if layer(state) >= r-1 then do
                begin
                    avoidArray := empty
                    for a := every symbol in A do
                        begin
                            if $i \in endposition(\delta(state, a))$ then
                                add a to avoidArray
                        end
                    if size of avoidArray $\neq$ m then do
                        begin
                            d[i] := a random symbol from A - avoidArray
                            state := $\delta(state, d[i])$
                            i := i+1
                        end
                end
            else
                begin
                    d[i] := a symbol randomly selected from A
                    state := $\delta(state, d[i])$
                    i := i+1
                end
        end
    enter(d) /*Output d as a mature detector and add d into the detector graph*/
end
```

Fig. 4. Algorithm C: The heuristic detector generating algorithm based on the self graph. This algorithm describes the flow of generating a new detector. The symbols in all position of a new detector are generated one by one, and randomly selected from a subset of the coding alphabet that does not result in matching with the self patterns stored in the self graph.

4 Comparative Experiments and Results

In this section, the comparative experiments are carried out to make comparisons between the G-NSA and traditional NSA in different coding spaces. For convenience, the traditional NSA is indicated by T-NSA in the following sections. The random algorithm and the heuristic algorithm for generating the detector set are also compared in the following experiments. In the following comparisons, the G-NSA with the random detector set generating algorithm is indicated with G-NSA-r, and the G-NSA with the heuristic detector set generating algorithm is indicated with G-NSA-h.

The experimental procedure is outlined as follows:

(1) Fix P_f (the possibility that the detector set fails to detect an anomaly string) to a given value.

(2) Compute P_m according to equation (1) [7, 8]

$$P_m \approx m^{-r}[(l-r)(m-1)/m+1] \tag{1}$$

(3) Compute N_R based on P_f and P_m using equation (2) [2]

$$N_R \approx {}^{-\ln P_f}\!\big/_{P_m} \tag{2}$$

(4) Repeat the following steps for 100 times independently.

a) Generate N_S random strings to make up the self set S.

b) T-NSA generates random strings as candidate detectors, until N_R mature detectors are found.
G-NSA-r generates random strings as candidate detectors until N_R mature detectors are found, and its detector graph is constructed.
G-NSA-h generates N_R mature detectors, and its detector graph is constructed.

c) Generate N_T random strings to make up a test set T.
Detecting T by the detector set of T-NSA, the detector graph of G-NSA-r, and the detector graph of G-NSA-h, respectively.

(5) Compute the statistic values of SpaceS, SpaceR, Candidates and TimeCost for T-NSA, G-NSA-r and G-NSA-h.

Four parameters, namely SpaceS, SpaceR, Candidates and TimeCost, are selected to make the comparison between T-NSA, G-NSA-r and G-NSA-h. These parameters are compared by their average values and standard deviations.

SpaceS: The space cost of the self set. For T-NSA, it is N_S*l. For G-NSA, it is the size of the self graph, which consists of two parts. The one is (m+1) times of the number of state nodes in the self graph. This is because each node in the self graph includes a *next move* function and a *layer* function. Note that the *next move* function stores the state transitions for all possible symbols with an array of size m, and the *layer* function stores the layer information of each node with one number only. The other part is the total size of the *endposition* arrays in the nodes at the last layer, and note that the lengths of the arrays can be different.

SpaceR: The space cost of the detector set. For T-NSA, it is N_R *l. For G-NSA, it is the size of the detector graph, which is calculated in the same way as the size of the self graph.

Candidates: The number of candidate detectors tried by the detector generating algorithm.

TimeCost: For T-NSA, *TimeCost* means the average number of detectors used by the detection algorithm for detecting a string. For the G-NSA-r or G-NSA-h, *TimeCost* means the average number of state transitions in the detector graph when applying the partial matching algorithm. In fact, for fair comparisons, because the partial matching operations are conducted between the detector and the string to be detected, the *TimeCost* of T-NSA should be the average number of matching operations performed by the detection algorithm. Since the *TimeCost* of G-NSA-r and G-NSA-h is much lower than T-NSA, as demonstrated in the following experimental results, only the average number of detectors used by the detection algorithm is adopted here.

4.1 Binary Space Experiments

In this subsection, the coding alphabet A={0 ,1}, and m is 2. The P_f is fixed as 0.05.

In the first experiment, the coding length l is set as 16, the matching threshold r is set as 12. Then the N_R is 4090 according to formula (2). The N_S is 1000, and the N_T is 60000. The average results over 100 independent runs are listed in Table 1. The standard deviations of the average values are also given in the table.

In the second experiment, the coding length l is set as 24, the r is set as 14. And consequently the N_R is 8180 according to formula (2). The N_S is set as 5000, and the N_T is 100000. The average results are listed in Table 2.

Table 1. The comparison results in binary space when l=16, r=12 and N_S=1000. The average values over 100 independent runs and the corresponding standard deviations of the four parameters are listed in this table.

		SpaceS	SpaceR	Candidates	TimeCost
T-NSA	Ave.	16000.00	65440.00	8510.31	1576.71
	Std.	0.00	0.00	125.00	15.63
G-NSA-r	Ave.	24814.23	34271.49	8474.19	13.29
	Std.	115.89	101.18	108.39	0.02
G-NSA-h	Ave.	24814.23	34534.20	4529.15	13.16
	Std.	115.89	89.92	27.12	0.02

Table 2. The comparison results in binary space when l=24, r=14 and N_S=5000. The average values over 100 independent runs and the corresponding standard deviations of the four parameters are listed in this table.

		SpaceS	SpaceR	Candidates	TimeCost
T-NSA	Ave.	120000.00	196320.00	51068.54	3013.53
	Std.	0.00	0.00	689.75	17.88
G-NSA-r	Ave.	143950.26	158055.26	51069.98	17.45
	Std.	177.67	233.99	724.30	0.03
G-NSA-h	Ave.	143950.26	161283.25	10760.84	17.06
	Std.	177.67	202.48	71.15	0.02

Table 1 and Table 2 show that, in the two experiments, the space costs for storing the self set of both G-NSA-r and G-NSA-h are greater than that of T-NSA, while the space costs for storing the detector set of both G-NSA-r and G-NSA-h are lower than that of T-NSA.

During the detector set generating process, the number of candidates tried by G-NSA-h is evidently less than those of T-NSA and G-NSA-r.

Compared to T-NSA, the most obvious improvement of G-NSA-r and G-NSA-h lies in the time cost when perform partial matching between an input string and the detector set. The time costs of G-NSA-r and G-NSA-h are less than *l*.

4.2 A-Z Space Experiment

In this subsection, an experiment is carried out in the alphabet of {A, B, …, Z}, and *m* is set as 26. As the A-Z coding space is far lager than the binary coding space, the coding length *l* is set as a smaller value 6 here, and *r* is set as 3. The P_f is fixed to 0.01, and the N_R is 20836 in accordance with formula (2). In this experiment, N_S=10000, N_T=100000. The average results over 100 independent runs are listed in Table 3.

Table 3. The results in the (A-Z) coding space when *l*=6, *r*=3 and N_S=10000. The average values over 100 independent runs and the corresponding standard deviations of the four parameters are listed in this table.

		SpaceS	SpaceR	Candidates	TimeCost
T-NSA	Ave.	60000.00	125016.00	189781.56	5742.28
	Std.	0.00	0.00	1916.68	25.83
G-NSA-r	Ave.	472605.71	491654.97	189749.24	3.92
	Std.	964.36	858.47	1745.00	0.01
G-NSA-h	Ave.	472605.71	486142.11	20836.00	3.95
	Std.	964.36	927.28	0.00	0.01

In the A-Z space experiment, Table 3 shows that the space costs of G-NSA-r and G-NSA-h are much greater than that of T-NSA. This is because the *next move* function for each node should store the 26 possible states. However, the number of candidates used by G-NSA-h is obviously less than those of both T-NSA and G-NSA-r. When perform partial matching between an input string and the detector set, the time costs of G-NSA-r and G-NSA-h are less than the coding length 6.

4.3 Classification Problem Experiments

In this subsection, two experiments are carried out to compare the T-NSA and the G-NSA with two classification problems. The Tic-Tac-Toe Endgame data set and the SPECT heart data set [9] are used as the classification data sets in the experiments. The experimental procedure applied in this subsection is the same as that in subsection 4.1 and 4.2.

4.3.1 Experiments on the Tic-Tac-Toe Data Set

The tic-tac-toe data set contains 958 instances, and these instances are partitioned into 2 classes, a positive class with 626 instances and a negative class with 332 instances. Every instance has 9 attributes in a 3-dimension space. Consequently, this subsection takes *A*={a, b, c}, *m*=3, and *l*=9.

In the following experiments, the negative class is considered as the self set, i.e. N_S= 332. The whole data set is used as the test set, i.e. N_T =958. In the test phase, all instances in the data set will be classified as self or non-self.

In the first experiment, the r is set to 6, the P_f is fixed to 0.01, and the N_R is set to 1119 by formula (2). The results of the first experiment are listed in Table 4. The classification results are also listed in Table 4 by the last column named "*MatchCount*". Here the column *MatchCount* lists the average numbers of the instances of the test set T matched by the detector set of T-NSA or the detector graph of G-NSA-r or G-NSA-h.

In the second experiment, the r is set to 7, the P_f is fixed to 0.01, and the N_R is set to 4316 by formula (2). The results of the first experiment are listed in Table 5.

In the third experiment, the r is set to 8, the P_f is fixed to 0.05, and the N_R is set to 11792 by formula (2). The results of the first experiment are listed in Table 6.

In these simple classification problem experiments, the comparison results are similar to the results of the experiments in subsection 4.1. The column *SpaceS* and *SpaceR* in Table 4, Table 5 and Table 6 reveal that the space costs for storing the self set of G-NSA-r and G-NSA-h are greater than that of T-NSA, while the space costs for storing the detector set of G-NSA-r and G-NSA-h are less than that of T-NSA.

Table 4. Taking the negative class of the tic-tac-toe data set as the self set, this experiment is performed with parameters m=3, l=9, r=6 and P_f=0.01. The whole data set is used as the test set. The average values over 100 independent runs and the corresponding standard deviations of the four parameters are listed in this table. Column MatchCount lists the average numbers of the positive instances of the test data set matched by the detector set of T-NSA or the detector graph of G-NSA.

		SpaceS	SpaceR	Candidates	TimeCost	MatchCount
T-NSA	Ave.	2988.00	10071.00	2415.40	618.73	538.62
	Std.	0.00	0.00	47.22	11.07	9.16
G-NSA-r	Ave.	3398.00	5060.21	2408.47	7.58	538.03
	Std.	0.00	38.67	52.93	0.03	7.93
G-NSA-h	Ave.	3398.00	5111.46	1300.40	7.52	565.99
	Std.	0.00	31.90	14.26	0.03	5.19

Table 5. Taking the negative class of the tic-tac-toe data set as the self set, this experiment is performed with parameters m=3, l=9, r=7 and P_f=0.01. The whole data set is used as the test set. The average values over 100 independent runs and the corresponding standard deviations of the four parameters are listed in this table. Column MatchCount lists the average numbers of the positive instances of the test date set matched by the detector set of T-NSA or the detector graph of G-NSA.

		SpaceS	SpaceR	Candidates	TimeCost	MatchCount
T-NSA	Ave.	2988.00	38844.00	5664.38	2217.54	588.35
	Std.	0.00	0.00	42.87	28.93	6.09
G-NSA-r	Ave.	5515.00	14383.87	5665.33	7.88	589.65
	Std.	0.00	44.99	44.66	0.02	4.90
G-NSA-h	Ave.	5515.00	14404.97	4423.29	7.84	612.17
	Std.	0.00	43.90	10.59	0.02	3.49

Table 6. Taking the negative class of the tic-tac-toe data set as the self set, this experiment is performed with parameters m=3, l=9, r=8 and P_f=0.05. The whole data set is used as the test set. The average values over 100 independent runs and the corresponding standard deviations of the four parameters are listed in this table. Column MatchCount lists the average numbers of the positive instances of the test date set matched by the detector set of T-NSA or the detector graph of G-NSA.

		SpaceS	SpaceR	Candidates	TimeCost	MatchCount
T-NSA	Ave.	2988.00	106128.00	12807.05	6425.49	598.19
	Std.	0.00	0.00	31.00	78.36	4.86
G-NSA-r	Ave.	6612.00	39933.35	12809.78	8.44	597.58
	Std.	0.00	66.85	32.21	0.01	5.62
G-NSA-h	Ave.	6612.00	39925.52	11844.39	8.45	595.62
	Std.	0.00	65.57	7.09	0.01	5.14

In the column *Candidates*, G-NSA-h used less candidate detectors than G-NSA-r and T-NSA to generate the detector set. The column *TimeCost* indicates that G-NSA-r and G-NSA-h require shorter time than T-NSA to process an input string.

Moreover, according to the column *MatchCount* in Table 4 and Table 5, the classification result of G-NSA-h is better than those of T-NSA and G-NSA-r.

4.3.2 Experiments on the SPECT Heart Data Set

The SPECT heart data set describes diagnosing of cardiac Single Proton Emission Computed Tomography (SPECT) images [9]. The data set consists of 267 instances of patients, and these instances are partitioned into 2 categories: normal (55 instances) and abnormal (212 instances). Every instance has 22 binary attributes. Therefore, this subsection takes A={a, b}, m=2, and l=22.

In the following experiments, the normal class is used as the self set, i.e. N_S= 55. The whole data set is used as the test set, i.e. N_T =267, the P_f is fixed to 0.05.

In the first experiment, the r is set as 12, and the N_R is 2045 by formula (2). The results of the first experiment are listed in Table 7. The classification results are also listed in the table by the last column named "*MatchCount*" just as in subsection 4.3.1.

In the second experiment, the r is set to 13, and the N_R is set as 4462 by formula (2). The results of the first experiment are listed in Table 8.

In the last experiment, the r is set to 14, and the N_R is set as 9816 by formula (2). The results of the first experiment are listed in Table 9.

In this subsection, the comparison results are also similar to the results in subsection 4.1. The column *SpaceS* and *SpaceR* in Table 7, Table 8 and Table 9 reveal that the space costs for storing the self set of G-NSA-r and G-NSA-h are greater than that of T-NSA, while the space costs for storing the detector set of G-NSA-r and G-NSA-h are less than that of T-NSA.

The column Candidates indicates that the time cost of G-NSA-h is a little lower than G-NSA-r and T-NSA in the detector set generating process. The column TimeCost shows that G-NSA-r and G-NSA-h have far lower time costs than that of T-NSA in processing an input string. And it can be also found out that the classification result of G-NSA-h is better than those of T-NSA and G-NSA-r in the three experiments for the SPECT heart data set.

Table 7. Taking the normal instances of the SPECT heart data set as the self set, this experiment is performed with parameters m=2, l=22, r=12 and P_f=0.05. The whole data set is used as the test set. The average values over 100 independent runs and the corresponding standard deviations of the four parameters are listed in this table. Column MatchCount lists the average numbers of the abnormal instances of the test data set matched by the detector set.

		SpaceS	SpaceR	Candidates	TimeCost	MatchCount
T-NSA	Ave.	1210.00	44990.00	2111.13	1150.47	167.66
	Std.	0.00	0.00	7.42	50.98	6.18
G-NSA-r	Ave.	2564.00	42165.29	2110.14	17.27	168.23
	Std.	0.00	78.22	9.15	0.27	7.00
G-NSA-h	Ave.	2564.00	42188.52	2048.75	17.11	174.33
	Std.	0.00	90.89	1.79	0.24	4.97

Table 8. Taking the normal instances of the SPECT heart data set as the self set, this experiment is performed with parameters m=2, l=22, r=13 and P_f=0.05. The whole data set is used as the test set. The average values over 100 independent runs and the corresponding standard deviations of the four parameters are listed in this table. Column MatchCount lists the average numbers of the abnormal instances of the test data set matched by the detector set.

		SpaceS	SpaceR	Candidates	TimeCost	MatchCount
T-NSA	Ave.	1210.00	98164.00	4532.12	2439.16	172.19
	Std.	0.00	0.00	8.22	96.89	4.58
G-NSA-r	Ave.	2930.00	83382.20	4530.99	17.55	172.73
	Std.	0.00	106.46	8.59	0.22	5.26
G-NSA-h	Ave.	2930.00	83398.81	4467.50	17.42	176.32
	Std.	0.00	131.49	2.39	0.17	3.64

Table 9. Taking the normal instances of the SPECT heart data set as the self set, this experiment is performed with parameters m=2, l=22, r=14 and P_f=0.05. The whole data set is used as the test set. The average values over 100 independent runs and the corresponding standard deviations of the four parameters are listed in this table. Column MatchCount lists the average numbers of the abnormal instances of the test data set matched by the detector set.

		SpaceS	SpaceR	Candidates	TimeCost	MatchCount
T-NSA	Ave.	1210.00	215952.00	9887.84	5272.57	174.41
	Std.	0.00	0.00	7.83	210.35	5.64
G-NSA-r	Ave.	3208.00	164357.73	9886.77	17.93	174.37
	Std.	0.00	159.57	9.89	0.17	4.30
G-NSA-h	Ave.	3208.00	164393.56	9818.41	17.85	177.52
	Std.	0.00	168.01	1.45	0.18	4.16

5 Discussions

Some works have been done to improve the performance of the traditional NSA. Helman and Forrest proposed a linear algorithm for generating antibody strings under the “r-contiguous-bits” matching rule [10]. Based on the work in [10], a greedy algorithm to generate smaller detector set with better coverage of non-self space was proposed by D'haeseleer and Forrest [11, 12]. A real-valued NSA with variable sized detectors was presented by Zhou and Dasgupata in [13, 14]. Zhang and his colleagues introduced a r-adjustable NSA in [15]. In [16], a novel NSA named as r[]-NSA was proposed, which uses an array to store multiple partial matching lengths for each detector. Aiming at improving the detection performance of NSA, a state graph approach to storing the self set and the detector set is proposed in this paper, and the “r-contiguous-bits” partial matching rule is adopted in the graph data structure.

According to theorems in [1], the time costs of constructing the functions of the pattern matching machine (i.e. the self graph or detector graph) is proportional to the sum of the lengths of the keywords (i.e. the self strings or detector strings).

The experimental results indicate that the time cost of G-NSA is much lower than T-NSA in processing an input string. In fact, when processing an input string, the G-NSA makes no more than l state transitions. Therefore, the time complexity of the G-NSA is O(l). However, the time complexity of the T-NSA is $O(N_R \cdot l)$.

In the T-NSA, the space used by all the symbols of the detector set is $N_R \cdot l$. Thus the space complexity of the T-NSA is $O(N_R \cdot l)$. G-NSA may have a larger space complexity that T-NSA. For example, Table 3 demonstrates that the space costs of both G-NSA-r and G-NSA-h are much greater than that of T-NSA, especially for *SpaceS*. This is because the *next move* function for each node should store the m possible state transitions. In addition, the self data used in Table 3 are generated randomly. So these self individuals often have no identical prefixes. When G-NSA is used for practical applications, and the self individuals have similar prefixes, the number of the total states in the self graph of G-NSA will decrease. And then the space cost of G-NSA will decrease somewhat. Anyway, this should be verified in the future works.

6 Conclusions

Negative Selection Algorithm is widely applied in Artificial Immune Systems. Taking the multi-pattern matching algorithm proposed by Aho and Corasick in 1975 [1] as an inspiration, a novel negative selection algorithm, i.e. G-NSA, is introduced in this paper. This algorithm constructs a state graph from strings of the self set or the detector set, and processes input strings using the “r-contiguous-bits” partial matching rule based on the state graph.

The G-NSA is compared with the traditional NSA in the experiments conducted in this paper. The results confirmed that the G-NSA has obviously lower time complexity when process an input string.

Further works should be done to improve the performance of G-NSA, especially the detector set generating algorithm based on the self graph. And more detailed comparisons should be done, including the comparative experiments and analyses between the G-NSA and the improved negative selection algorithms in [10, 12-16].

Acknowledgements. This work is partially supported by the National Natural Science Foundation of China (NO. 60404004), and the open foundation from Anhui Key Laboratory of Software in Computing and Communication.

References

1. Aho, A.V., Corasick, M.J.: Efficient String Matching: An Aid to Bibliographic Search. Communications of the ACM 18(6), 333–340 (1975)
2. Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R.: Self-Nonself Discrimination in a Computer. In: Proceedings of 1994 IEEE Symposium on Research in Security and Privacy, pp. 202–212. IEEE Computer Society Press, Los Alamitos, CA (1994)
3. Wu, S., Manber, U.: Agrep - a Fast Approximate Pattern Matching Tool. In: Proceedings of USENIX Winter Technical Conference, pp. 153–162 (1992)
4. Wu, S., Manber, U.: A Fast Algorithm for Multi-Pattern Searching. Technical Report, TR 94-17. University of Arizona, Tucson, AZ (May 1994)
5. Wang, Y., Shen, Z., Xu, Y.: Improved Algorithms for Matching Multiple Patterns (in Chinese). Journal of Computer Research and Development 39(1), 55–60 (2002)
6. Tuck, N., Sherwood, T., Calder, B., Varghese, G.: Deterministic Memory Efficient String Matching Algorithms for Intrusion Detection. In: Proceedings of IEEE Infocom 2004, Piscataway, pp. 333–340. IEEE Computer Society Press, Los Alamitos (2004)
7. Percus, J.K., Percus, O.E., Perelson, A.S.: Predicting the Size of the T-Cell Receptor and Antibody Combining Region from Consideration of Efficient Self-Nonself Discrimination. Proceedings of the National Academy of Sciences 90, 1691–1695 (1993)
8. Percus, J.K., Percus, O.E., Perelson, A.S.: Probability of Self-Nonself Discrimination. In: Perelson, A.S., Weisbuch, G. (eds.) Theoretical and Experimental Insights into Immunologyedn, pp. 63–70. Springer, New York (1993)
9. Kurgan, L., Cios, K., Tadeusiewicz, R., Ogiela, M., Goodenday, L.: Knowledge Discovery Approach to Automated Cardiac Spect Diagnosis. Artificial Intelligence in Medicine 23(2), 149–169 (2001)
10. Helman, P., Forrest, S.: An Efficient Algorithm for Generating Random Antibody Strings, Technical Report CS-94-07. Department of Computer Science, University of New Mexico, Albuquerque, New Mexico (1994)
11. D'haeseleer, P.: Further Efficient Algorithms for Generating Antibody Strings, Technical Report CS95-3. Department of Computer Science, University of New Mexico, Albuquerque, New Mexico (1995)
12. D'haeseleer, P., Forrest, S., Helman, P.: An Immunological Approach to Change Detection: Algorithms, Analysis and Implications. In: Proceedings of 1996 IEEE Symposium on Security and Privacy, pp. 110–119. IEEE Computer Society Press, Los Alamitos, CA (1996)
13. Zhou, J., Dasgupata, 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)
14. Zhou, J., Dasgupata, D.: Augmented Negative Selection Algorithm with Variable-Coverage Detectors. In: Proceedings of the 2004 Congress on Evolutionary Computation (CEC '04), vol. 1, pp. 1081–1088 (2004)
15. Zhang, H., Wu, L., Zhang, Y., Zeng, Q.: An Algorithm of R-Adjustable Negative Selection Algorithm and Its Simulation Analysis (in Chinese). Chinese Journal of Computers 28(10), 1614–1619 (2005)
16. Luo, W., Wang, X., Tan, Y., Wang, X.: A Novel Negative Selection Algorithm with an Array of Partial Matching Lengths for Each Detector. In: Runarsson, T.P., Beyer, H.-G., Burke, E., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) Parallel Problem Solving from Nature - PPSN IX. LNCS, vol. 4193, pp. 112–121. Springer, Heidelberg (2006)

Clonal Selection Algorithms for 6-DOF PID Control of Autonomous Underwater Vehicles

Jongan Lee[1], Mootaek Roh[1], Jinsung Lee[2], and Doheon Lee[1]

[1] Department of Bio and Brain Engineering
[2] Robotics Program,
Korea Advanced Institute of Science and Technology(KAIST), Republic of Korea
{jalee,mtroh,jslee,dhlee}@biosoft.kaist.ac.kr

Abstract. Autonomous underwater vehicles(AUVs) have been drawing increasing interests in various marine applications such as coastal structure inspection, sea floor exploration, and oceanographic monitoring. Due to the complexity of underwater stream dynamics and the prevalence of unexpected underwater obstacles, it is imperative to develop self-adjustable, intelligent navigation control functions for AUVs. Among various control techniques, we focus on the proportional-integral-derivative(PID) controller since it is still one of the dominant techniques in actual underwater vehicle control systems. We propose to apply the Clonal Selection Algorithm to determine optimal combination of three gain coefficients, K_P, K_D, K_I of the PID controller. Our simulation shows that the proposed technique provides better responses than the existing Ziegler-Nichols technique with respect to the settling time, overshoot and an affinity in submerging under water and turning the yaw angle through simulation. We expect that AUVs could autonomously regulate three coefficients of six degree-of-freedom(DOF) PID controllers through real-time onboard processing.

Keywords: Autonomous Underwater Vehicles, 6-DOF PID controller, Clonal Selection Algorithms (CSA), Ziegler-Nichols technique.

1 Introduction

Autonomous underwater vehicles(AUVs) have become an important tool for various underwater tasks because they have greater speed, endurance, and depth capability as well as a higher factor of safety than human divers. However, most vehicle control system designs have been based on a simplified vehicle model, which has often resulted in poor performance because the nonlinear and time-varying vehicle dynamics have coefficient uncertainty. It is desirable to have an advanced control system with the capability of learning and adapting to changes in the vehicle dynamics and parameters [1]. Thus, AUVs need the autonomous coefficient tuning due to the complexity of underwater stream dynamics and the prevalence of unexpected underwater obstacles. There have been a few advanced control techniques of AUVs. Autonomous diving and steering of unmanned underwater vehicles can be controlled by multivariable sliding mode control [2]. Discrete-time Quasi-sliding

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 182–190, 2007.

mode systems have been adapted for control of autonomous underwater vehicles [3]. Recently, the revised precision controller is tested through model parameters optimization using the Nelder-Mead Simplex Technique [4]. Although a few control techniques have been suggested, they have some difficulties to apply autonomous underwater vehicles. We utilize the classical control technique which is the PID controller to determine the attitude and position of AUVs. They have been successfully applied to many different problems in control fields and have achieved valuable results [5]. Basically, the classical Ziegler-Nichols technique has been used for tuning PID controllers [6]. The modified Ziegler-Nichols technique has been suggested to improve the performance and efficiency of the classical Ziegler-Nichols technique [7]. Recently, the PID neural network is used for the temperature control system [8]. The purpose of this work is to ascertain the effect of using Clonal Selection Algorithms for 6-DOF PID controllers of AUVs.

2 Dynamic for 6-DOF of Autonomous Underwater Vehicles

For Autonomous Underwater Vehicles(AUVs) in 6 Degrees of Freedom(DOF) the dynamic equations of motion are usually separated into the translational and rotational motions. The position is specified by three vectors which are surge, sway and heave. On the other hand various representations of an attitude have been discussed. Between them, the most frequently applied representations of it are Euler angle conventions which are minimal three parameter representations. The roll, pitch and yaw convention dominate in the context of mobile vehicles [9]. There are significant coupling problems between the rotational and translational motion for the 6 DOF underwater vehicles control. Therefore, several mathematical models have been proposed to solve these problems. Among them, we used underwater robotic vehicle dynamics model proposed by T. I. Fossen, 1994 [10]. It contains kinematic equations of motion, rigid-body dynamics, added inertia, hydrodynamic damping, and restoring forces.

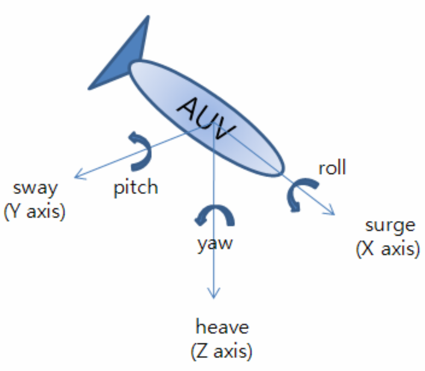


Fig. 1. Body fixed coordinate system and 6-DOF of AUVs

3 6-DOF PID Controllers of AUVs Applied Clonal Selection Algorithms

3.1 PID Controllers

A proportional-integral-derivative(PID) controller is a general feedback loop component in control systems. Each coefficient of a PID controller works three common requirements of control problems. The proportional part works to handle an immediate error, the error is multiplied by a constant(K_P). The integral part works to learn from the past, the error is integrated and multiplied by a constant(K_I). The derivative part works to predict the future, the first derivative is calculated and multiplied by a constant(K_D). The PID controller can change outputs adequately based on the history and rate of change of the error, which gives more accurate and stable systems.

Table 1. The effect of increasing coefficients

	Rise time	**Overshoot**	**Settling time**
K_P	Decrease	Increase	Small change
K_I	Decrease	Increase	Increase
K_D	Small change	Decrease	Decrease

3.2 Rigid Body Dynamics of AUVs

Newton's equations of the motion for rigid-body with constant mass are written [11]:

$$m[\dot{u} - vr + wq - x_G(q^2 + r^2) + y_G(pq - \dot{r}) + z_G(pr + \dot{q})] = \sum X . \qquad (1)$$

$$m[\dot{v} - ur + wp - y_G(r^2 + p^2) + z_G(qr - \dot{p}) + x_G(qp + \dot{r})] = \sum Y . \qquad (2)$$

$$m[\dot{w} - wq + vp - z_G(p^2 + q^2) + y_{xG}(rp - \dot{q}) + y_G(rq + \dot{p})] = \sum Z . \qquad (3)$$

$$I_x \dot{p} + (I_x - I_y)qr - mz_G(\dot{v} + ur - wp) = \sum K . \qquad (4)$$

$$I_x \dot{q} + (I_x - I_z)pr + m[z_G(\dot{u} - vr + wp) - x_G(\dot{w} - wq + vp)] = \sum M . \qquad (5)$$

$$I_x \dot{r} + (I_y - I_x)pq + mx_G(\dot{v} - wp + ur) = \sum N . \qquad (6)$$

where x_G, y_G, z_G is the center of gravity, m is the constant mass, I_x, I_y, I_z is the inertia matrix of AUVs, $\upsilon^B = [u, v, w]^T$ is the velocity of the origin of body axis relative to fluid, $\omega^B = [p, q, r]^T$ is the angular velocity component about each axis relative to fluid, X, Y, Z and K, M, N are vectors of external applied forces and moments, respectively.

3.3 6-DOF PID Controllers for AUVs

The closed-loop system is guaranteed by using standard stability analysis [12]. The error is calculated as the difference between the actual distance, Euler angle and target distance, Euler angle.

$$\tilde{\varepsilon} = \varepsilon_a - \varepsilon_T . \tag{7}$$

$$s = K_P \tilde{\varepsilon} + K_D \dot{\tilde{\varepsilon}} + K_I \int_{t_0}^{t} \tilde{\varepsilon}(\tau) d\tau . \tag{8}$$

where K_P, K_D, K_I are gain coefficients and $\varepsilon_a, \varepsilon_T$ is the actual distance, Euler angle and the target distance, Euler angle. According to gain coefficients, the system responds differently. We calculate moderate K_P, K_D, K_I gain coefficients by using the classical Ziegler-Nichols technique.

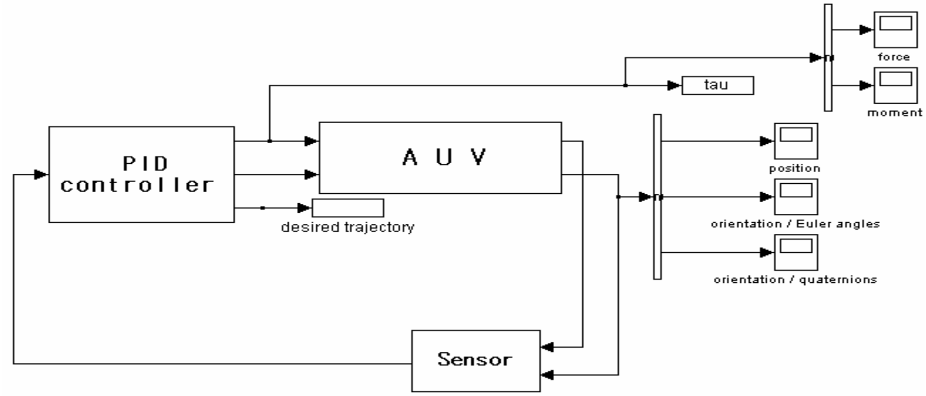


Fig. 2. Block diagram of the 6-DOF PID controller

3.4 Clonal Selection Algorithms for Tuning PID Controller of AUVs

The immune system is capable of learning, memory, and pattern recognition. By employing genetic operators on a time scale fast enough to observe experimentally, the immune system is able to recognize novel shapes without preprogramming [13].

Artificial immune systems(AIS) use ideas gleaned from immunology in order to develop adaptive systems capable of performing a wide range of tasks in various areas of research. The general algorithm, named CLONALG, is derived primarily to perform machine-learning and pattern-recognition tasks and then it is adapted to solve optimization problems, emphasizing multimodal and combinatorial optimization [13]. The Clonal Selection Algorithm(CSA) is based on the artificial immune system. The CSA is used in the field of optimization and pattern recognition [15].

The movement of AUVs can be optimally controlled by selecting its PID gain coefficients. Usually, the gain coefficients for the PID controller are measured by empirical experiments. Therefore, it is appropriate to apply the CSA to get the optimal gain coefficients. First, generate a population of PID gain values randomly and compute all the affinity values for that. The affinity measures are computed by the equation (9). After that, the select gain values with high affinity. And then, they are identically copied, and mutated with high rates. These are replaced with the gain values in the initial population which have lower affinity. These processes are iteratively performed until CSA gives converged, optimal gains. Finally, all PID gain coefficients are optimally selected by CSA. This allows the control system of AUVs to operate without high-overshoot or longer-settling time.

$$Affinity = \frac{1}{1 + error} . \tag{9}$$

$$Error = \sum_{i=1}^{n} \left| A_i - T_i \right| . \tag{10}$$

A_i : The actual distance and Euler angle against each axis(x, y, z),

T_i : The target distance and Euler angle against each axis(x, y, z)

4 Simulation Position and Attitude of AUVs

We determined some parameters and surroundings for simulation.

- **Weight of the AUVs : 120 kg**
- **Shape of the AUVs : Sphere type with six thrusters for 6 DOF**
- **Buoyancy : Neutral / Place : An indoor swimming pool**
- **Target depth : 5 meters, Target yaw angle : 90 degree**
- **Population size : 50 / Clone per antibody : 6**
- **Generation : 30 / Number of bits : 16**
- **Hypermutation rate : 0.05**
- **Simulation time : 120(s) for depth control, 30(s) for yaw angle control**

4.1 Experiment 1. The Depth Control of AUVs

The first simulation is to keep 5 meters under water. We do not need to consider an external force because we assume the place an indoor swimming pool. Results are compared to the classical Z-N technique.

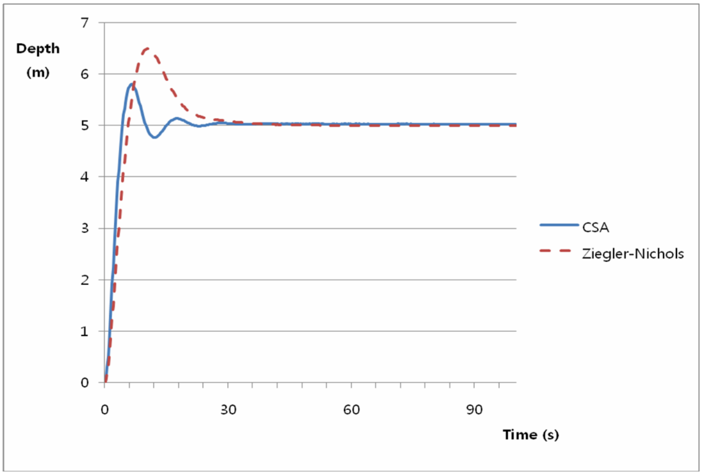


Fig. 3. The depth control of AUVs between CSA and classical Z-N technique

We know that proportional coefficient K_P and derivative coefficient K_D are larger than the integral coefficient K_I. Although the CSA has larger K_P coefficient, the overshoot is less than the classical Z-N technique.

Table 2. The comparison of PID controller efficiency for depth control of AUVs

	Classical Ziegler-Nichols	Clonal Selection Algorithm
K_P	41.7	65.53
K_I	4.04	0.12
K_D	107.37	65.50
Maximum overshoot	6.5 (m)	5.8 (m)
settling time	28.4 (s)	19.2 (s)
affinity	0.0064	0.0110

4.2 Experiment 2. The Yaw Angle Control of AUVs

The second simulation is turning 90 degrees the yaw angle while AUVs keep the depth under water. Similar to the experiment 1, we assume that the external force does not exist.

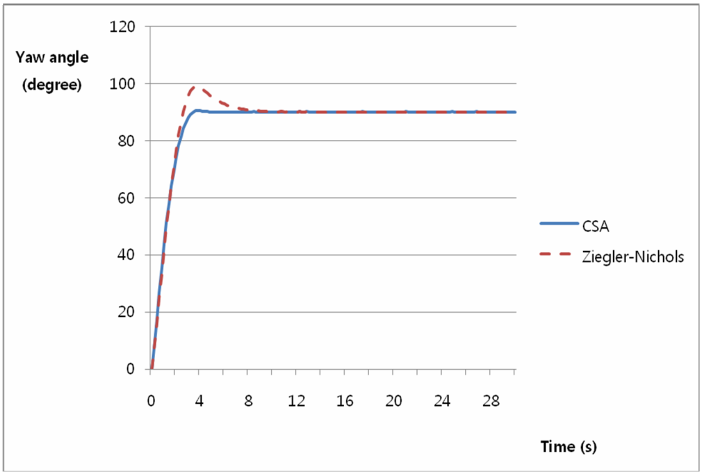


Fig. 4. The yaw angle control of AUVs between CSA and classical Z-N technique

Compared to the experiment 1, the settling time is reduced. It means that the PID controller has the different gain according to the 6 DOF of the AUVs. Thus, we have to check each coefficient K_P, K_D, K_I through the previous simulation.

Table 3. The comparison of PID controller efficiency for yaw angle control of AUVs

	Classical Ziegler-Nichols	Clonal Selection Algorithm
K_P	61	65.53
K_I	43.5	0.018
K_D	21.57	0.005
Maximum overshoot	99.1 (degree)	90.1 (degree)
settling time	7.8 (s)	4.8 (s)
affinity	0.0014	0.0017

5 Discussion

In the experiment 1, we easily find that the settling time of the CSA is shorter than the classical Z-N technique. The overshoot of the CSA is less than the classical Z-N technique. As the proportional coefficient K_P and the integral coefficient K_I increase, the overshoot of the response is decreased. However, the overshoot of the response is increased as the derivative coefficient K_D increase. This means that each coefficient of the PID controller gain K_P, K_I, K_D is mutually affected to the response. In addition, the affinity of the CSA is approximately two times comparing to the classical Z-N technique. As a result, we know that CSA is the more efficient technique and guarantees the excellent performance. In addition, results of the experiment 2 are similar comparing to those of the experiment 1. The settling time of CSA is shorter than the classical Z-N technique and the overshoot of CSA is less than the classical Z-N technique. On the other hand, the affinity of the CSA is similar to the Z-N technique.

We assume the place an indoor swimming pool. However, AUVs are utilized to navigate under sea. They could encounter unpredictable situations and be affected the density of water and the current along the depth of water. Thus, they should easily change PID coefficients along situations to enhance performance. AUVs have the threshold of settling time, overshoot and affinity of PID controller. If they exceed the threshold value, AUVs could recalculate PID coefficients based on their trace.

We should consider many factors in developing AUVs with mechanical and electrical views. For example, the external hull is important because buoyancy is different along with the body shape. After AUVs are made, we test them through the simulation to maximize the performance and efficiency of AUVs. First of all, attitude and position control of AUVs are important to execute their missions. Thus, we could get a valuable improvement of AUVs through simulation.

6 Conclusion

It is important to keep the body of AUVs stable during missions. Several technicians have suggested control systems for AUVs control. Among them, we adapt the classical control technique PID controller. Our present study evaluates the performance and efficiency of 6-DOF PID controllers through the CSA comparing to the classical Z-N technique. We verify that the CSA is more efficient than the Z-N technique in submerging and turning yaw angle through the simulation. We expect that AUVs could autonomously regulate three coefficients of six degree-of-freedom (DOF) PID controllers through real-time onboard processing for undersea exploration.

Acknowledgements

This research was supported by the Ministry of Information and Communication(MIC), Korea, under the Information Technology Research Center(ITRC) support program supervised by the Institute of Information Technology Advancement(IITA) (IITA-2006-C1090-0602-0001). We would like to thank CHUNG Moon Soul Center for BioInformation and BioElectronics for providing research facilities.

References

1. Yuh, J.: Learning Control for Underwater Robotic Vehicles. IEEE Journal of Oceanic Engineering 18(3) (1993)
2. Healey, A.J., Lienard, D.: Multivariable Sliding Mode Control for Autonomous Diving and Steering of Unmanned Underwater Vehicles. IEEE Journal of Oceanic Engineering 18(3) (1993)

3. Lee, P.M., Hong, S.W., lim, Y.K., Lee, C.M., Jeon, B.H., Park, J.W.: Discrete-Time Quasi-Sliding Mode Control of an Autonomous Underwater Vehicle. IEEE Journal of Oceanic Engineering 24(3), 388–395 (1999)
4. Mark, E.R., Franz, S.H: System Identification of Open-Loop Maneuvers Leads to Improved AUV Flight Performance. IEEE Journal of Oceanic Engineering 31(1) (2006)
5. Astrom, K.J., Hagglund, T.: PID Controller: Theory Design and Tuning. Instrument Society of America (1995)
6. Ziegler, J.G, Nichols, N.B.: Optimal Settings for Automatic Controllers. Trans. ASME 64(11), 759–768 (1942)
7. Astrom, K.J., Hagglund, T.: PID Controllers, 2nd edn. ISA, N. C (1995)
8. Huailin, S., Youguo, P.: Decoupled Temperature Control System Based on PID Neural Network. In: ACSE 05 Conf. vol. 5, pp. 107–111 (2005)
9. Fjellstad, O.E., Fossen, T.I.: Position and Attitude Tracking of AUVs: A Quaternion Feedback Approach. IEEE Journal of Oceanic Engineering 19(4), 512–518 (1994)
10. Fjellstad, O.E., Fossen, T.I: Singularity-Free Tracking of Unmanned Underwater Vehicles in 6 DOF. IEEE Conference on Decision and Control, 1128–1133 (1994)
11. Gertler, M., Hagen, G.R.: Standard Equation of Motion for Submarine Simulations, NSRDC Report No. 2510 (1967)
12. Khalil, H.K.: Nonlinear Systems. Macmillan, NYC (1992)
13. Farmer, J.D., Packard, N.H.: The immune system, adaptation, and machine learning. Physica D: Nonlinear Phenomena 22(1), 187–204 (1986)
14. Castro, L.N., Zuben, F.J.: Learning and optimization using the Clonal Selection Principle. IEEE Trans. on Evolutionary Computation 6(3) (2002)
15. Castro, L.N., Zuben, F.J.: The Clonal Selection Algorithm with Engineering Applications. In: Workshop Proceeding of GECCO, pp. 36–37 (2000)

An Immuno Robotic System for Humanitarian Search and Rescue
(Application Stream)

Henry Y.K. Lau and Albert Ko

Intelligent Systems Laboratory
The University of Hong Kong
Pokfulam Road, Hong Kong SAR
aux1496@gmail.hku.hk

Abstract. 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. To fully realize the promising characteristics of robotics search and rescue systems, however, mobile robots must first be free from their tether and be granted the ability to navigate autonomously even when wireless control commands from the operator cannot reach them. For search and rescue robots to go autonomous in exceedingly unstructured environment, the control system must be highly adaptive and robust to handle all exceptional situations.

This paper introduces the control of a low-cost robotic search and rescue system based on an immuno control framework, GSCF, which was developed under the inspiration of the suppression mechanism of the immune discrimination theory. The robotic system can navigate autonomously into rubbles and to search for living human body heat using its thermal array sensor. Design and development of the physical prototype and the control system are described in this paper.

Keywords: Artificial Immune Systems, Emergency Logistics, Humanitarian Search and Rescue, USAR, Robotics.

1 Introduction

Humanitarian search and rescue operations can be found in most large-scale emergency operations. Search and rescue technology to-date rely on search dogs, camera mounted probes, and technology that has been in service for decades. With the increasing demand for scapegoats to go into dangerous environment to carry out reconnaissance and into hazardous environments to perform inspection, robots are being identified as good candidates to step in for their creator – human. Robots

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 191–203, 2007.

equipped with advanced sensors have therefore become more and more popular in the search and rescue theatre. To navigate autonomously, search and rescue robots require robust and adaptive fail-soft systems that can ensure calculable reliability desired for operations under unstructured environment.

Human immune system is a robust and adaptive decentralized system; the function of its components and their interactions offer inspiring analogies for solving problems in different disciplines. General Suppression Control Framework (GSCF) based on the suppression mechanism between immune cells was designed to take advantage of the adaptability and robustness of its biological counter part. This research demonstrates the possibility to implement GSCF on a decentralized search and rescue robot system. The goals of the research is to design and develop a decentralized control system based on GSCF to assist a search and rescue robot system to communicate and to navigate in unstructured disaster-affected areas.

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 mechatronic design of the newly developed robot prototype and its control strategies. An introduction to AIS and the implementation of GSCF into the new search and rescue system is also included in the second half of the paper. Recommendations and conclusions is given 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 [Centers for Disease Control and Prevention 1995], the Oklahoma Tornado in 1999 [National Severe Storms Laboratory 1999], the Indian Ocean Earthquake [Zubair 2004] and Hurricane Katrina in 2005 [Federal Emergency Management Agency 2005], and the Pakistan Earthquake in 2005 [Birsel 2005], 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 Figure 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.

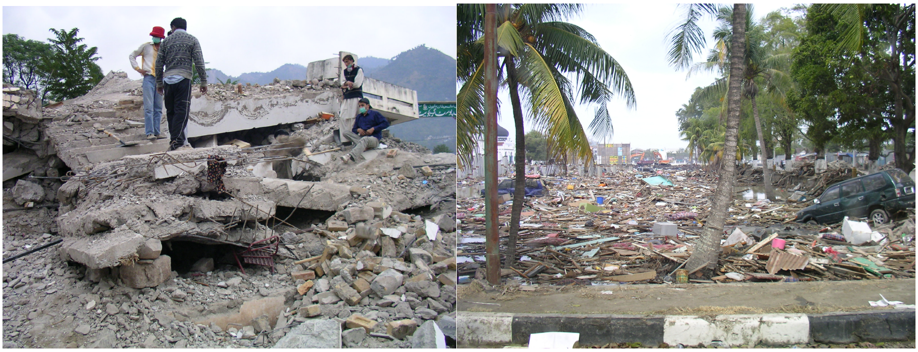


Fig. 1. Left: Pakistan earthquake 2005, locals making attempts to search for survivors in a collapsed girl's college. The structure was in unstable condition; excavation and lifting machineries were prohibited from the site. **Right:** Indian Ocean Earthquake 2004. Most buildings were collapsed and roads were blocked by debris. (Pictures taken on site by author during the two relief missions).

3 Search and Rescue Robots

Mobile robots designed for search and rescue operations are rugged in design and offer many features to address current technology constraints. Search and rescue robots can navigate through voids and crevices that are too small for search dogs, and can zigzag between obstacles to reach areas where straight camera mounted probes cannot reach. Search and rescue robots equipped with camera and two way voice communications allows the operator to get a visual image of the victim's surrounding and to speak to the victim to provide psychological support. Moreover, once the location of the victim is identified, another robot can deliver water and food to prolong the victim's life.

Robots for search and rescue had been discussed in scientific literature since the early 1980's [Kobayashi and Nakamura 1983]; 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 [Murphy 2004a]. 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 [Snyder 2001]. Their primary functions are to search for victims and paths through the rubble that would be quicker to excavate, perform structural inspection and detection of hazardous material.

4 Mechatronic Design

At present, search and rescue robots are typically stand-alone unit focused in obstacle avoidance and object discovery. In real world operations, when these robots are deployed into rubbles to search for victims or into muddled mine fields to locate bombs, they are often deterred by narrow passages. Smaller robots can often reach deeper into rubble pile than larger, bulkier robots; however, smaller robots are more likely to report lost if they drop themselves into large openings. Since search and rescue robots are typically deployed to work in highly unstructured environment, damaging and losing of robots due to uncontrollable external factors should not be considered as failures; instead, all loses should be considered as calculable risk and incorporate the risk into normal operation cost. Following this line of thought, to minimize operation cost, small low-cost search and rescue robots that can be deployed in high volume are more suitable for search and rescue missions in unstructured environment than large complex single unit.

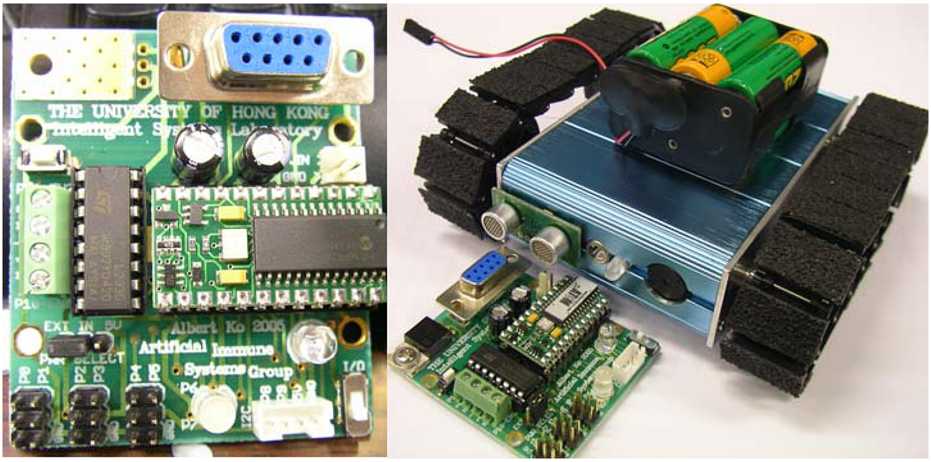

THE UNIVERSITY OF HONG KONG
Intelligent System Laboratory
Albert Ko 2005
Artificial Immune Systems Group

Fig. 2. Left: Custom designed control board for general AIS robot controls. **Right:** The physical prototype of the newly developed robot. The battery pack on top of the robot serves as a scale to show the robots' dimension.

The search and rescue robot system being discussed in this paper consists of an operation console and two autonomous robots. The robots are essentially two mechatronically loaded aluminum cases, each fitted with two tread-belts driven by two separate gear-motors. Each of these robots is equipped with a Thermal Array Sensor (TAS), a camera for transmitting visual image to the operator, a microphone for picking up sound under the rubble, an accelerometer to tell the orientation of the robot in respect to gravitational pull, a sonar range finder for obstacle avoidance, a high intensity LED for lighting, 6V rechargeable battery, a custom designed micro controller board designed for general AIS robot controls, and a ZigBee wireless network module for establishing a network between the operator and the robots. The board has two channels for motors up to 2A, 6 servo controllers, one 5V regulator, two LED indicators, one I2C port and a serial port for programming (Figure 2). The prototype has the camera, TAS, and high-intensity LED encased in the aluminum case. Sonar is to be encased in the second prototype.

The operation console consists of a mini-monitor for displaying video images obtained from the robots, a ZigBee wireless network module for communication and a remote control unit for interfacing human inputs to the mechatronic system. Basic layout of the system is illustrated in Figure 3.

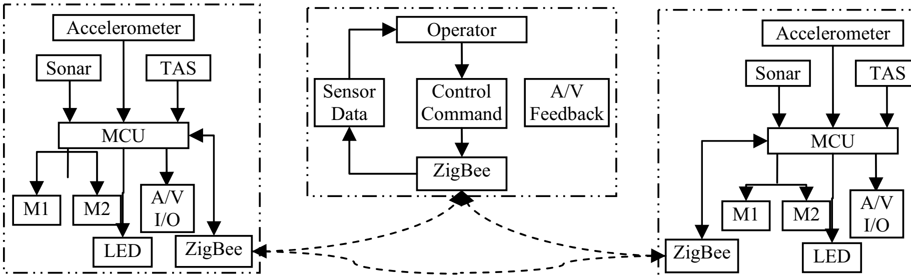


Fig. 3. Mechatronic layout of the system

The designated function of the robots is to navigate autonomously into rubble to search for living bodies using the TAS equipped in front of the robot. The TAS is a thermopile array that detects infrared in the 2μm to 22μm range. The unit has eight thermopiles arranged in a row and can measure the temperature of 8 adjacent points simultaneously. These thermopiles are identical to those used in non-contact infrared thermometers, and can detect heat generated from a human body from 2 meters away regardless of the lighting condition. The robots can avoid obstacles and find passage under rubble autonomously using its sonar range finder, but the operator can, at anytime, choose to control each robot individually using the remote controller with the assistant of the control console's mini-monitor. This alternative control scheme enables the human operator to assist the robot to solve navigation problems based on real-time visual images. Without this alternative control scheme, the robots would require many more onboard sensors and higher computational power to achieve barely comparable results, which in turn adds to the weight and cost of the robots.

The robots are designed with two separate communication channels to minimize power consumption. The A/V channel for audio and video uses 2.4GHz transmission, and is by default turned off to save power. When the robot detects an object with human-body-like temperature or when it has difficulty to navigate out of a trap, it will generate a request for the operator to turn on the A/V channel and assist it to navigate or to determine if the object is a human being. The data channel is for sharing data and transmitting command between robots and the operator. This is done through the second channel using the ZigBee (http://www.freescale.com/ZigBee) communication modules installed in the robots and in the control console. ZigBee is a low-power, short-distance wireless standard based on 802.15.4 that was developed to address the needs of wireless sensing and control 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. This unique feature is highly desirable for search and rescue robots operating in unstructured environment. The ZigBee communication channel can also be turned off to save power, and can be waken wirelessly with a single command. In fact, it is programmed to stay in standby mode when it is not transmitting or receiving data.

To effectively achieve the designated function, the robots are instructed to behave in two distinct modes in respond to external stimulations. These two distinct modes govern the robots' actions in victim searching and in exception handling. When the robot is behaving in search mode, it uses its sonar to identify open passages and navigates autonomously into the rubble to look for possible victims using its TAS. While in this mode, the robot shuts down all onboard devices that are not directly related to its objective to conserve energy for navigation and exploration. In practice, the A/V system and the high intensity LED for illumination are deactivated under exploration mode. When the robot identifies a possible victim based on data obtained from the TAS, or when the robot believes it has trapped itself in the rubble, it will switch to the exception-handling mode to request for operator assistance. While in exception-handling mode, the robot would first send all data related to its current situation (i.e. the most current set of data from TAS and sonar) plus its current status (i.e. possible victim identified or trapped) to the operation console. Then it shuts down all energy consuming devices, put the ZigBee communication module to standby mode and wait for the operator's assistance. The human operator can reactivate the robot wirelessly by responding to the console. Once the robot is reactivated in exception-handling mode, it would reinitiate the A/V device, the LED, the sonar, the TAS, and the motor controllers to assist the human operator to determine whether the object identified is a living human body. The human operator can also remotely control the robot to navigate out of a trap with the assistant of the video feedback. The robot can switch back to exploration mode at the operator's command. External interruptions (operator commands or help requests) received through ZigBee communication module can also cause the robot to enter exception-handling mode.

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) [de Castro and Timmis 2003] 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. Though AIS is still in its infancy when compared to other well-established computational intelligence paradigms such as, evolutionary algorithms, artificial neural network and fuzzy systems; its promising underlying biological principles has attracted many researchers from different field.

Scientists and engineers have applied AIS to solve a wide variety of problem. [Lau & Wong 2004] developed a control framework to improve the efficiency of a distributed material handling system. [de Castro & Timmis 2002] presented the application of AIS in computer network security, machine learning, and pattern recognition. [Sahan et. al 2005] applied attribute weighted AIS to diagnosis heart and diabetes diseases. [Dasgupta et al. 2004] exploited negative selection algorithm to detect abnormalities in aircrafts. [Cserey et al. 2004] developed an AIS real-time visual analysis system for surveillance based on the behavior of T-cells. [Oda & White 2005] 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, [Ko et al. 2004a] developed a General Suppression Control Framework (GSCF) for designing control systems for modular robots based on the suppression mechanism in AIS. The framework has also been applied to design control systems for controlling a two-wheeled self-balancing robot in heterogeneous-connected mode [Ko et al. 2005]. This paper, continuing from previous works, describes the application of GSCF in designing robust decentralized control systems for a small platoon of search and rescue robots.

6 General Suppression Control Framework

The General Suppression Control Framework (GSCF) [Ko et al. 2005a] is based around the analogy of the immunological suppression hypothesis in the discrimination theory [Aickelin et al. 2003]. 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) [Sharon 1998] 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 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.

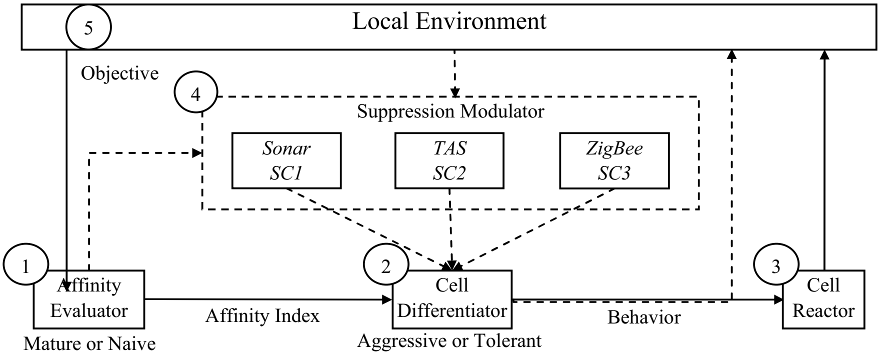


Fig. 4. The General Suppression Control Framework. Dashed lines represent humoral signal transmissions, where solid lines represent cellular signals. The suppression modulator can host any number of suppressor cells.

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 Figure 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 Control Design and System Integrations

The search and rescue robot control system presenting in this paper is based on the GSCF [Ko et al. 2004b] developed for controlling decentralized systems. In general, the first step in designing a GSCF based control system is to identify the system objective and system constraints. For the search and rescue robots in this research, the primary objective is to search for human body under rubbles using their TAS. Therefore searching for human-body-temperature-like heating object is the system objective. Next, for the robot to navigate through rubbles to search for heat, the robots must be able to avoid obstacles and to ask for help when it is stuck. Therefore avoiding obstacle is a crucial condition that the robots must satisfy before pursuing the system objective; hence obstacle avoidance is a system constraint. In addition, operator commands and help requests, made by other robots within the system, received through ZigBee communication module are also treated as external constraints.

With system objective and constraints identified, the next step is to organize these conditions into system solvable form. 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 search and rescue robot, the Affinity Evaluator is responsible for monitoring the status of the system objective. The system objective is said to have achieved when a human-body-temperature-like heating object is detected. The Affinity Evaluator would produce a high affinity index when the system object is achieved to encourage the system to behave aggressively. When a robot is in aggressive mode, it would remain in its position and perform a series of actions to alarm the operator for assistant. Otherwise, the Affinity Evaluator would produce a low affinity index to allow the system to continue exploring the surrounding to search for heating objects. When the affinity index is low, 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 search and rescue robots under discussion have two main sensors that determine the robots' behaviors. The sonar range finder helps the robot to avoid obstacles, and the TAS helps to locate heating objects. Suppressor cells that have high sensitivity to these sensors are situated in the Suppression Modulator.

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. The function of Cell Differentiator, on the other hand, 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. Similar to Suppression Modulator, Cell Differentiator is also an important component of GSCF; it 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 Figure 5.

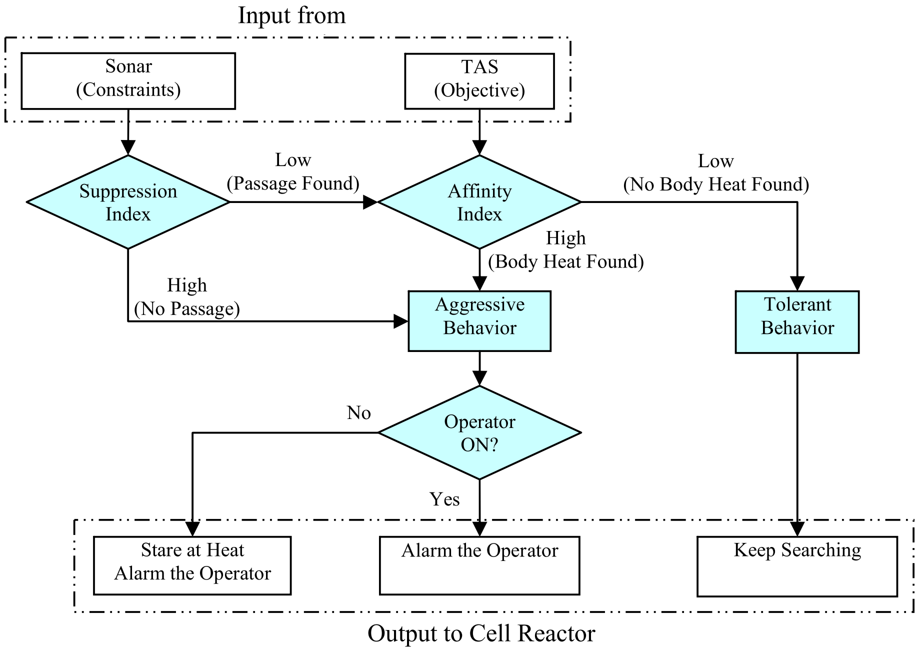


Fig. 5. Decision scheme in the Cell Differentiator of each modular fireguard

The suppression indices from the suppressor cells have priority over all others, it is being evaluated first to see whether the robot is blocked by obstacles or has found a heating body. If the suppression index is high, meaning the system has detected something unusual; the suppressor modulator can force the robot to behave in aggressive mode instantly. On the other hand, when the suppression index is low, the system will check the affinity index and follow the normal procedures to determine how the robot should behave.

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 Observation Test

To evaluate the performance of a field system and to determine points of improvement, the GSCF-based search and rescue robot system is put to test in a semi-unstructured environment. The test environment is a dumpsite for old furniture and equipment; the piling of chairs, broken pallets, and construction debris resembles a condition close to an earthquake-affected indoor environment. The purpose of this test is to observe how GSCF handles the robots' different behaviors based on simple suppression mechanism. The robots ability to transform between aggressive and tolerant behavior in response to the external condition is also of the experiment's main focus.

The robot deployed into the test environment performed as designed. The robot navigates autonomously into the rubble to search for heat emitting objects that are close to the temperature of heat emitted from a living human body. The robot stopped and switched into tolerant mode after it detected the operators hand. The operator then took over the control of the robot, navigated it to a different location, and let the robot resume its patrolling. Mobility of the robot is biased towards certain terrain. The small size of the robot inherently handicapped its mobility over terrains with large holes, as the robots would simply fall through them as it strolls over. When in front of narrow passages, the robot demonstrated good mobility. The equipped accelerometer helped the robot to determine if it is flipped over and allow the control system to change its motor directions accordingly, so the robot can continue to move in the same direction after being flipped in an accidental event. This feature proves to be very useful over rough terrain and narrow passages, as the operator does not need to know which side of the robot is up to drive the robot forward.

Since this is only a prototype for testing the concept of controlling low-cost autonomous search and rescue robots with GSCF based system, it is fair to say the performance of the robot is inline with design expectation and the GSCF based control system works well as the backbone of the system. To further develop the current prototype system, certain improvements can be made. Suggestions and recommendations derived from this research are discussed in the concluding section.

9 Recommendations and Conclusions

This paper presented a low-cost search and rescue robot system that can navigate into voids in rubbles, avoid obstacles, detect living human body temperature, transfer video image, and communicate in a low-power ZigBee network. The robot system consists of two robots and one operator console and can be expended to consist any number of robots. The GSCF based control system enables the system to be controlled in decentralized manner using very simple commands and limited communication power.

In spite of the technological challenges and mistrust of new technologies in human nature, search and rescue robots will become an indispensable tool in future rescue operations. Starting to develop and field search and rescue robots with regular rescue teams can help scientists to better understand the strength and weaknesses of different robot designs under different situations. Having robots working in parallel with regular rescue team can also help scientists to investigate how robots should behave to comply with their operators' instructions and to best assist the rescue effort in general.

Simple user control interface allows amateur rescuers to be trained to operate the robot in a short period of time, eliminating the need to occupy limited professionals to look after each robot. Low manufacturing cost allows robots to be deployed in mass quantity to increase the chance of finding survivors. Battery is the heart of robots; it keeps electricity pumping inside the robots. Lighter, smaller and more powerful battery is also an important constituent of effective search and rescue robots. Emergency wireless network for communication is also important for coordinating actions between robots, collecting visual image from the robots, and to communicate with the victim when the robot finds one.

References

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)

Birsel, R.: Pakistan: Frantic Search as Pakistani Quake Toll Tops 20,000. ReliefWeb. Retrieved 12th June, 2006, from the World Wide Web (October 10, 2005), http:// www.library.cornell.edu/newhelp/res_strategy/citing/apa.html#apa

Centers for Disease Control and Prevention. Deaths Associated With Hurricanes Marilyn and Opal – United States, September-October 1995. Retrieved June 27, 2006, from the World Wide Web (January 19, 1996), http:// www.cdc.gov/ mmwr/ preview/ mmwrhtml/00040000.htm

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)

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)

de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, New York (2002)

de Castro, L.N., Timmis, J.: Artificial Immune Systems as a Novel Soft Computing Paradigm. Softing Computing 7(8), 526–544 (2003)

Federal Emergency Management Agency. Cash Sought to Help Hurricane Victims, Volunteers Should Not Self-Dispatch. Retrieved June 27, 2006, from the World Wide Web (August 29, 2005), http://www.fema.gov/news/newsrelease.fema?id=18473

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 (IROS2004), Sendai, Japan, pp. 1774–1779 (2004a)

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)

Ko, A., Lau, H.Y.K., Lau, T.L.: An Immuno Control Framework for Decentralized Mechatronic Control. International Journal of Unconventional Computing, 225–280 (2005a) (to appear)

Kobayashi, A., Nakamura, K.: Rescue Robot for Fire Hazards. In: Proc. of International Conference on Advanced Robotics, pp. 91–98 (1983)

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)

Murphy, R.R.: National Science Foundation Summer Field Institute for rescue robots for research and response (R4). AI Magazine 25(2), 133–136 (2004a)

National Severe Storms Laboratory. May 3rd, 1999 Oklahoma/Kansas Tornado Outbreak. Retrieved 22th May, 2006, from the World Wide Web (May 3, 1999), http:// www.nssl.noaa.gov/headlines/outbreak.shtml

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)

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)

Sharon, J.: Basic Immunology. Williams & Wilkins, Pennsylvania, USA (1998)

Snyder, R.: Robots assist in search and rescue efforts at WTC. IEEE Robot. Automation Magazine 8, 26–28 (2001)

Zubair, L.: Scientific Background on the Indian Ocean Earthquake and Tsunami. Retrieved June 27, 2006, from The International research Institute for Climate and Society (December 28, 2004), http:// www.library.cornell.edu/ newhelp/ res_strategy/ citing/ apa.html#apa

The Application of a Dendritic Cell Algorithm to a Robotic Classifier

Robert Oates, Julie Greensmith, Uwe Aickelin, Jonathan Garibaldi, and Graham Kendall

The University of Nottingham
{rxo,jqg,uxa,jmg,gxk}@cs.nott.ac.uk
http://www.asap.cs.nott.ac.uk

Abstract. The dendritic cell algorithm is an immune-inspired technique for processing time-dependant data. Here we propose it as a possible solution for a robotic classification problem. The dendritic cell algorithm is implemented on a real robot and an investigation is performed into the effects of varying the migration threshold median for the cell population. The algorithm performs well on a classification task with very little tuning. Ways of extending the implementation to allow it to be used as a classifier within the field of robotic security are suggested.

1 Introduction

Technologies and protocols designed to enforce security are now pervasive in society. Most houses now have burglar alarms, CCTV is common-place in towns and cities and the private security industry is estimated to provide products and services up to the value of £4 billion in the UK alone [1]. It is possible to group most existing solutions as either 'manned guarding' or static-sensor networks.

Manned guarding (bouncers, private security guards etc.) is a popular technique for providing additional security to buildings containing expensive or sensitive items. Human security systems are difficult to pre-empt and can adapt to new circumstances. However, human performance varies greatly and is heavily reliant on rest periods. People are also susceptible to prejudices and preferences depending on gender, race and age. Guarding is a potentially hazardous occupation as it places an individual between a criminal and their goal.

Static sensor networks (CCTV, standard burglar alarms etc.), can be stored and replayed as and when required. They do not require rest and react predictably to all situations. If damaged, static sensors are easy to replace and in systems with centralised data storage, evidence is not compromised. However, criminals can plan around static sensors; as they can be obscured and cannot negotiate obstacles. Static sensor networks cannot effectively use short-range sensor-types, unless deployed in bottle-necks, such as entry and exit points. The limiting factor for many static sensor networks is the volume of information generated. Very few sensors can be monitored by an individual effectively. Tickner *et al.* estimated that the number of feeds that a single operator can effectively

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 204–215, 2007.

monitor is approximately 16, with the detection rate falling from 83% for a four camera system, to 64% for a 16 camera system, [2].

Robotic systems have many properties to make them a useful tool for security applications. They have the advantages of static sensor networks and are capable of moving around obstructions to gain a better line of sight. Short range sensors are more effective when mounted on a robot, as the sensor can be taken to the target. Whilst it is possible for an automated sentry to become predictable, intelligent routing algorithms could make evasion challenging. The key disadvantage of a robotic system is the volume of data. The camera mounted to the front of a robot is likely to be even more difficult to monitor than a static camera, as both the background, and items of interest will be moving on the screen. This disadvantage could potentially be overcome if the robots could autonomously recognise events of interest and report them to the operator.

Artificial immune systems (AIS) have had numerous successes in the field of anomaly detection. A newly developed AIS algorithm, the dendritic cell algorithm (DCA), is a promising technique for the processing of time-dependant data [3]. The DCA is based on recent developments in immunology regarding the role of dendritic cells (DCs), as a major control component within the immune system. The DCA is based on an abstraction of DC behaviour and performs fusion of data from disparate sources. Successful applications of the DCA have focussed on solving intrusion detection problems in computer security, a field which shares properties with problems in both robotics and physical security.

The potential benefits of applying the DCA to a robotic security solution are numerous. It is hoped that the DCA will have a resilience to the noise associated with real-world signals due to its ability to fuse information from disparate sources via a population of artificial cells. The aim of this investigation is to explore the applicability of the DCA to a robotic system. In section 2 we present work relevant to the areas of security robotics and the DCA. In section 3 the implementation of a general robotic DCA is discussed. Section 4 outlines an investigation into the effects on the performance of altering the dendritic cell migration threshold when applied to a trivial robotic classification problem. The results of this investigation are presented, analysed and discussed in sections 4.1 to 4.6. In section 5 conclusions are drawn about the applicability of the algorithm to robotic security and possible extensions of this work are outlined.

2 Related Work

2.1 Robotic Systems

Developing a robotic system is a demanding task as robust, real-time control is difficult to achieve. Brooks' "subsumption architecture" (first proposed in [4]), has been shown to be an effective way of designing robotic control systems [5]. Such architectures rely on the development of a family of simplistic "behavioural modules" that interact to produce more complex behaviour. For example, the complex behaviour of wandering through a dynamic environment, without hitting obstacles can be achieved through the interaction of two simplistic behavioural modules. Figure 1 illustrates a simple subsumption architecture. The lower priority behaviour simply moves the robot forwards at a constant velocity. In the event of a higher priority behaviour detecting an obstacle, it can subsume the output of the low priority behaviour and steer the robot away or, in emergencies, stop. The interaction between the two modules ensures that the robot is always moving when possible, without hitting obstacles.

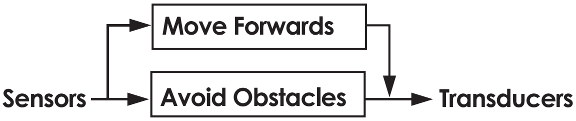


Fig. 1. A simple subsumption architecture for implementing a wandering behaviour

2.2 Autonomous Security Systems

Whilst the robotic security problem is yet to be rigorously formalised, an architecture using robots as autonomous scouts which report 'interesting' events to a human operator has precedent [6][7][8]. Using this approach, the robotic security problem can be viewed as two, well-researched problems: path planning and classification. Massios *et al.* [9] define patrol route planning as an optimisation problem, minimising the probability of missing a "relevant event".

The classification problem is the discrimination of important events from normal events. The "mobile detection assessment and response system" [8] is an American military project aimed at producing a collection of robots for interior and exterior security. These systems use basic motion-detection algorithms on data from an on-board camera [7]. The movement detection algorithm is simplified by keeping the robot base stationary during classification. When movement is detected other sensors are employed in conjunction with the camera to assess if the observed object is human or not. In [6], a more intelligent classification technique is proposed using colour analysis and clustering to compare a room's current state with its previously observed state. This algorithm has applications for identifying erroneous objects, e.g. unattended luggage, and recognising the theft of objects that were present in the test image.

2.3 The Dendritic Cell Algorithm

The DCA was conceptualised and developed by Greensmith *et al.* [10]. The algorithm is based on the behaviour of DCs, which are the antigen presenting cells of the immune system. DCs are natural anomaly detectors and data fusion agents, responsible for controlling and directing appropriate immune responses. The fusion of 'signals' across a population of DCs and the asynchronous correlation of signals with 'antigen' provides the basis of the DCA's classification. DCs exist in one of three states, immature, semi-mature and mature. Immature DCs perform signal fusion and process antigen. Semi-mature and mature DCs present antigen with a context value derived from the fused signals. Antigen presented by semi-mature DCs are 'normal' and the antigen presented by mature DCs are 'anomalous'. The biological theory is beyond the scope of this paper, but the interested reader can refer to [10] and [11] for the relevant immunological details. A formal description of the DCA is provided in [3].

The DCA is a population-based algorithm, with each agent in the system represented as a cell. Each cell can collect data items to classify forming antigen for use within the DCA. The DCA used in this paper relies on a '3-signal' model where three categories of input signal are used to produce three output signals. Signals and antigen are read into a signal matrix and antigen vectors. Antigen is sampled by DCs and removed from the tissue antigen vector and transferred to the DC's own antigen storage facility. Once antigen is sampled, the DC copies the values of the tissue signal matrix to its own signal matrix. These values are processed by the DC during the update to form cumulative output signal values. Equation 1 is the function used to process the signals, where O are output signals, S are input signals, i is the number of output signals, j is the number of input signals and W_{ij} is the weight used for O_i and S_j.

$$O_i = \sum_{j=1}^{3} W_{ij} S_j \qquad \forall i \tag{1}$$

Input and output signals are termed after their biological counterparts:

PAMPs (S_1): A signature of abnormal behaviour e.g. number of errors per second. This signal is proportional to confidence of abnormality.

Danger Signal (S_2): A measure of an attribute which increases in value to indicate an abnormality e.g. an increase in the rate of a monitored attribute. Low values of this signal may not be anomalous, giving a high value a moderate confidence of indicating abnormality.

Safe Signal (S_3): A measure which increases value in conjunction with observed normal behaviour e.g. a high value of S_3 is generated if the standard deviation of a monitored attribute is low. This is a confident indicator of normal, predictable or steady-state system behaviour. This signal is used to counteract the effects of PAMPs and danger signals and is assigned a negative weight in the weighted sum.

CSM (O_1): The costimulatory signal which is increased as a result of high values of all input signals. This value is used to limit the duration spent by DCs in the data sampling stage.

IL-10 (O_2): This value is increased upon the receipt of the safe signal alone.

IL-12 (O_3): This value is increased upon the receipt of PAMP and danger signals, and is decreased by the safe signal.

The processing of signals and antigen is distributed across the DC population to correlate disparate data sources to perform the classification of the algorithm. The DCA does not perform antigen pattern matching, unlike other AIS algorithms which perform antigen classification through analysis of the the structure of an antigen. Instead, the signals received by a DC during its antigen collection phase are used to derive an *antigen context* which is used to perform the basis of classification. This algorithm can be applied to problems where multiple antigens of identical structure i.e. antigens of the same type, are to be classified, such as the classification of anomalous processes [12].

Each DC is randomly assigned a *migration threshold value* which is compared against the cumulative $O1$ value. The details of the migration threshold value generation for this experiment can be found in section 4.4. If the value of $O1$ exceeds the migration threshold, the DC is removed from data sampling and enters the maturation stage. At this point the values for output signals $O2$ and $O3$ are assessed. If $O2 > O3$, the DC is termed 'semi-mature'. Antigen 'presented' by a semi-mature cell is assigned a context value of 0. Conversely, if $O2 < O3$ the cell is termed 'mature' and antigen presented by this cell is assigned a context value of 1. Once the DC has presented its antigen-plus-context values, it is reset and returned to the DC population. Data sampling by DCs continues for the duration of the experiment, or until a specified stopping condition is met.

After a specified number of antigen are presented by the DCs, analysis is performed. As mentioned, antigen do not have unique values representing their structure, with antigen of identical values termed as a 'type'. The MCAV coefficient (mature context antigen value) is calculated as the fraction of antigen presented in the mature context, per type of antigen. MCAVs close to 1 indicate that a type of antigen is potentially anomalous. A threshold is applied to the MCAV values to discriminate between anomalous and normal types of antigen.

Thus far, the majority of problems presented to the DCA are related to computer security, specifically the detection of port scans [3] it is also applied to a static machine learning dataset [10] and the detection of intrusions in sensor networks[13]. Work performed by Greensmith *et al.* [12] has indicated that the DCA performs well for time-dependent real-valued data, such as that seen in robotics applications. Following the interesting ideas proposed in [14], it is fathomable that the DCA, also based on innate immunity, could be incorporated into the field of mobile robotics. As the DCA has a history of good performance for illegal scan detection in computer security, it may be a useful algorithm for the purpose of physical robotic security applications.

3 The Robotic Dendritic Cell Algorithm

The DCA is applied to a general robotics problem to support the suitability of the algorithm for mobile robotic security. The platform used for this investigation is a Pioneer 3DX. This robotic system has a broad variety of sensors, including a laser range finder (LRF), an array of sonar sensors and a pan-tilt-zoom camera. Onboard processing is performed using an 850MHz Pentium III processor running Debian Linux, (kernel version 2.6.10). The manufacturer's "Aria" library is used to control the device. The Aria control system is an object-orientated (C++)

library which is structured to support the implementation of subsumption control architectures. All compilation was carried out using g++ version 4.0.2.

The robotic DCA is implemented as a stand-alone behavioural module for compatibility with a subsumption architecture. Figure 2 illustrates the architecture which implements a simultaneous wandering and DCA classifying behaviour. This extension of the Aria library's 'wander' architecture has an additional module for image processing and an additional module for executing the DCA. By making these additions part of the subsumption architecture, the fundamental behaviour of moving around safely within the environment can be prioritised above all other actions. In addition to the wandering and classifying behaviour, there is also a tele-operation, (remotely controlling the robot from a networked machine) and classifying behaviour. The DCA module outputs MCAV coefficients (as described in section 2.3), approximately once per second.

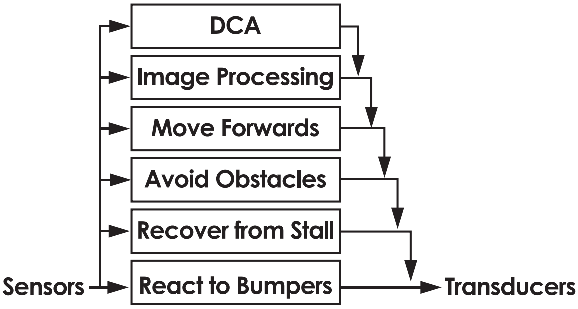


Fig. 2. The subsumption architecture used to implement the robotic DCA

The DCA used on the robot is a streamlined version of the algorithm which does not require any additional software libraries, unlike the implementation used in [12]. Verification of the streamlined implementation's functionality against that of the original DCA has been achieved. This was performed by attaching 'virtual' signals and antigen to the inputs of the module and processing the data used in [12]. The signal weightings specified for the anomaly detection algorithm in [12] were used for all experiments.

4 Experimental Validation

It is thought that the DCA is capable of processing real-time sensor data. It is further hypothesised that the migration threshold will have a noticeable effect on the false positive rate for this classification task. The following experiment is designed to test these ideas.

4.1 Experimentation

This experiment uses the “simultaneous wander and classify” behaviour discussed in section 3. The DCA classifies its current location as either ‘anomalous’ or ‘normal’ from the application-specific input signals. For this simple test, pink coloured objects with a height less than 330mm are considered anomalous whilst other obstacles are considered to be normal. The colour pink is used as it is easily distinguished from other objects within the robot’s environment. A height of 330mm is used as objects below that height are unobservable by the LRF’s planar field of view (FOV), but can still be detected by the sonar sensors’ conic FOV. This means objects classified as ‘pink’, detectable by the sonar but not detectable by the laser, are classified as anomalous.

The starting conditions for the experiments are illustrated by figure 3. Obstacle A is a pink cylinder, with a height less than 330mm which is an example of an anomalous object. Obstacle B is a pink cylinder, with a height greater than 330mm which is an example of a normal object. It is expected that the DCA will not react to the taller cylinder as the algorithm will prevent full maturation of the cells. By maintaining the starting position of the robot and the positions of the obstacles, it is possible to calculate the ideal classification for all points within the enclosure. The error between the theoretical response and the algorithm output can then be used as a metric to assess the performance of the algorithm.

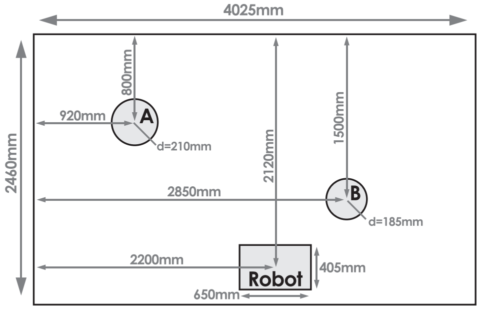


Fig. 3. The starting conditions for each experiment. Cylinder A is the ‘dangerous’ obstacle, cylinder B is the ‘safe’ obstacle.

4.2 Signal Sources

As described in section 2.3, three signals are used as inputs to the DCA inclusive of a safe signal, a danger signal and a PAMP signal. The former acts to suppress the full maturation of the dendritic cells, whilst the other two stimulate the maturation. All signals contribute to the migration of the cells.

The PAMP is sourced from the image processing module. The input from the camera is transferred into the HSV, (Hue, Saturation, Value) colour space and the histogram back-projection algorithm is applied to the data [15]. The back-projection algorithm uses a single training image to identify the colour properties of an object of interest. All pixel groups within the image that share the same statistical properties are identified and contours are drawn around those clusters. The final output from the image processing library is the area of the largest region which matches the properties of the test image. Intel's "OpenCV" library was used to perform all image processing. The output from the image processing module is scaled down before being used as the PAMP signal. The scaling factor used was calculated from test data generated by a seven minute random walk around the pen.

The LRF is used as the source for the safe signal so objects taller than 330mm will produce high values of the inhibitory signal. The FOV of the LRF extends from -90° to +90° (where 0° is directly in front of the robot). A 44° FOV is used, ranging from -22° to +22°. A narrow FOV reduces the risk of erroneous classification from walls. The distance to the closest object within the safe FOV is returned to the signal processor. The signal processor calculates the magnitude of the safe signal. This is performed using a look up table which relates distance to signal strength. For values that lie between those specified in the look up table, linear interpolation is used to calculate the signal strength. The values used are given in Table 1.

Table 1. Object Distance and Signal Strength for Ranged Sensors

Distance (mm)	Safe Signal Strength
0	100
300	90
600	50
900	20
1200	0

The danger signal is sourced from the sonar array which has a 360° FOV. The danger signal FOV coincides with the safe signal FOV. The same look up table (see Table 1) used to normalise the laser output for the safe signal is used to normalise the sonar output for the danger signal.

4.3 Antigen Source

In a practical robotic security solution, the antigen could be a vector based on the estimated location of the anomalous situation. Object-based approaches for antigen generation within a robot system have been put forward by by Krautmacher *et al.* in [16]. For this simple implementation antigen is an integer number which uniquely identifies a segment of the test pen. This encapsulates a small range of positions and orientations of the robot. The actual position and orientation of the robot is estimated using a 'dead reckoning' algorithm. Dead reckoning estimates the position and orientation of the robot from encoders mounted on the wheels, the fixed starting position of the robot and the diameter of the tyres.

The antigen generated enumerates a 300mm grid square within the pen and a 30° segment within that square. Generating antigen based on the current location of the robot is more practical than object-based antigen, which requires a deeper knowledge of the environment to compute.

As the antigen is generated based on a specific robot location, it is possible that an ineffective amount of antigen will be generated. One solution for this is to add multiple copies of each antigen to the DCA environment as suggested in [17]. A novel extension to the DCA for this application is an antigen multiplication function. This function adds varying amounts of each antigen depending on the speed of the robot. Areas passed through slowly are made to contribute more antigen than areas passed through quickly. This is done because areas passed through quickly contribute less signal to the DCA environment, as less time is physically spent within that area. The weighting function is given in equation 2.

$$W(v,\dot{\theta}) = 75\left(1-\left|\frac{v}{v_{max}}\right|\right)+1+25\left(1-\left|\frac{\dot{\theta}}{\dot{\theta}_{max}}\right|\right)+1 \tag{2}$$

In equation 2 v is the velocity of the robot, $\dot{\theta}$ is the rotational velocity of the robot and W is the amount of antigen added to the environment.

The smallest amount of antigen that can be added is 2, when the robot is at maximum velocity and maximum rotational velocity. The maximum amount of antigen that can be added is 102, when the robot is totally stationary.

A simple program written in Java calculates the theoretical MCAVs for every antigen, from the properties of the test pen. This is an unrealistic mathematical model of the experiment but provides a way of analysing the true and false positive rates for each run of the experiment.

4.4 Experiment Parameters

Each run of the experiment allowed the robot to wander around the test pen for ten minutes. The classification experiment was repeated three times for each value of migration threshold median. The experiments used the migration medians 15, 30, 60, 120 and 240. The range of allowed values is ±50% of the migration median in each case. Each DC was assigned a random migration threshold within the specified range, using an equi-probable distribution.

A naming convention is used referring to the first experiment as M15, the second as M30 etc. A threshold of 0.6 is applied to the MCAV values from the algorithm. Values less than or equal to 0.6 are counted as a negative or 'safe' classification, values above are counted as a positive or 'dangerous' classification.

4.5 Results

Figure 4 shows the false positive and false negative rates from the experiments. The rates are calculated by comparing the classification from the algorithm with the theoretical classification. Each point on the chart shows the misclassified antigen rate from the beginning of the experiment up to the time indicated on the x-axis. Each series is the average classification error from three runs with the specified migration rate.

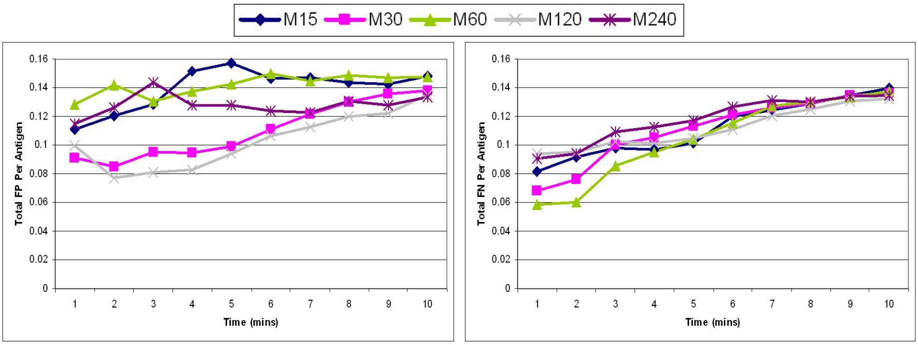


Fig. 4. The classification error rates from the experiments. The false positive rate is shown on the left and the false negative rate is shown on the right.

4.6 Analysis

The classification error rates rise throughout the experiments. Analysis of the robot's telemetry showed that the error in localisation from the dead-reckoning algorithm was drifting over time. As the measure used to assess the robot's performance relies upon the location of the robot, it is theorized that the classification errors from the first 1-2 minutes are closer to the "true" classification errors, as they will not be as significantly affected by the localisation drift. The use of a theoretical model as a baseline for the experiment could also introduce a constant error offset as the model may not be totally accurate. However, the performance of the algorithm is still high. The highest recorded rate of classification error for the entire experiment is a 0.16 false-positive rate, for the M240 experiment. The higher amount of antigen absorbed before migration increases the occurrence of cases when a DC collects both dangerous and safe antigen, making attributing 'blame' more difficult. The false positive rates all start below 0.14. M30 demonstrated the best performance overall, and appears to give the optimum performance for this particular experiment. It is intuitive to see higher error rates from experiment M15 as a low migration threshold will cause DC's to migrate after only sampling a small amount of signal. This would result in the classifier being more prone to noise within the system. M120 has amongst the highest false negative rates and the lowest false positive rates. More work will have to be done to understand why this should be the case. One potential cause may be the high range of possible migration thresholds with a tolerance of ±60. M60 yields the lowest false negative rate, but one of the highest false positive rates. The rates of error presented in 4 are lower than expected for this problem, indicating that the DCA is suitable for some robotic applications.

5 Conclusions

The misclassifications caused by the dead reckoning errors lead to the results being difficult to judge against the chosen metric. This could be a problem for future applications, as antigen generation for this application is intrinsically location specific. It is proposed that this issue could be resolved by adding a more advanced localisation algorithm based on using sensor readings to compensate for the integration errors.

It has been shown that it is possible to implement the DCA on a real robotic system. Whilst the problem was trivial, the low false positive and false negative rates are promising, especially considering that very little tuning or training has been performed. The implementation did not require any processing to be shared by another machine, so the DCA is scalable for an n-robot system and is usable in circumstances when the robot enters a region with poor communications coverage.

The next intended step for this project is to apply the DCA to a harder classification problem and compare its results with a fuzzy or neural classifier. This will provide an insight into the general performance of the DCA as a robotic classifier. Extending this work to a security system will require two key steps. Firstly the DCA will need to be modified to handle vector antigen instead of integer antigen. This will allow a more extendible representation of the environment to be used by the classifier. Secondly, the signal sources for a security system will need to be more complex than those used for this experiment. A possible source for the PAMP signal would be the error from a trained, non-linear model, correlating physical position to a normal scenario. Large error rates would imply an anomalous situation. The safe and danger signals could be controlled by the robot's physical location and the time of day. It would be advantageous to make the robot less sensitive during office hours and around busy public areas and more sensitive out of office hours and around high-security areas. A more generic anomaly detection system could be achieved through the introduction of a training-data based algorithm. The error rate between what the trained system expects to see and what its current sensor readings tell it, could be used as a source for PAMP signals.

Ultimately a multi-robot system, with dynamically changing routes and shared anomaly information could be developed, each using a DCA to assess the threat level for a given location.

Acknowledgements

Many thanks to William Wilson for his input to the software architecture and to Daniel Bardsley for his advice on image processing. The authors are very grateful to Mark Hammonds for generating the vector graphics for this paper. This work is financially supported by MobileRobots Inc.

References

1. SIA: The security industry authority annual report and accounts (2005-2006), Available at http://www.the-sia.org.uk/
2. Tickner, A.H., Poulton, E.: Monitoring up to 16 synthetic television pictures showing a great deal of movement. Ergonomics 14(4) (1973)
3. 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)
4. Brooks, R.A.: A robust layered control system for a mobile robot. IEEE J. Robotics and Automation, 14–23 (1986)
5. Brooks, R.A.: Elephants don't play chess. Robotics and Autonomous Systems 6, 3–15 (1990)
6. Castelnovi, M., Miozzo, M., Scalzo, A., Piaggio, M., Sgorbissa, A., Zaccaria, R.: Surveillance robotics: analysing scenes by colours analysis and clustering. In: CIRA (2003)
7. Everett, H., Gilbreath, G., Heath-Pastore, T., Laird, R.: Controlling multiple security robots in a warehouse environment. In: AIAA/NASA Conference on Intelligent Robots (1994)
8. Pastore, T., Everett, H., Bonner, K.: Mobile robots for outdoor security applications. In: ANS'99 (1999)
9. Massios, N., Voorbraak, F.: Hierarchical decision-theoretic robotic surveillance. In: IJCAI'99 Workshop on Reasoning with Uncertainty in Robot Navigation (1999)
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, Springer, Heidelberg (2005)
11. Lutz, M., Schuler, G.: Immature, semi-mature and fully mature dendritic cells: which signals induce tolerance or immunity? Trends in Immunology 23(9) (2002)
12. Greensmith, J., Twycross, J., Aickelin, U.: Dendritic cells for anomaly detection. In: Congress on Evolutionary Computation (CEC) (2006)
13. Kim, J., Bentley, P.J., Hailes, C.W.M.A.: Danger is ubiquitous: Detecting misbehaving nodes in sensor networks using the dendritic cell algorithm. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, Springer, Heidelberg (2006)
14. Neal, M., Feyereisl, J., Rascuna, R., Wang, X.: 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, Springer, Heidelberg (2006)
15. Swain, M., Ballard, D.: Color indexing. International Journal of Computer Vision 7(1) (1991)
16. 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)
17. Twycross, J., Aickelin, U.: Libtissue - implementing innate immunity. In: Congress on Evolutionary Computation (CEC'06) (2006)

On Immune Inspired Homeostasis for Electronic Systems

Nick D. Owens[1], Jon Timmis[1,2], Andrew J. Greensted[1], and Andy M. Tyrell[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. Many electronic systems would benefit from the inclusion of self-regulatory mechanisms. We strive to build systems that can predict, or be aware of, imminent threats upon their specified operation. Then, based on this prediction, the system can alter its operation or configuration to circumvent the effects of the threat. In this position paper, we discuss the role of the immune system can play in serving as inspiration for the development of homeostatic engineered systems, through the development of an immune inspired extensible architecture. We outline the major requirements for such an architecture, and discuss issues that arise as a result and propose possible solutions: things are never as simple as they first appear.

1 Introduction

The biological term Homeostasis coined by Cannon [1] refers to an organisms ability to maintain steady states of operation in a massively changing internal and external environment. Engineering homeostasis in electronic systems is a challenging endeavour. There have been many attempts at employing various mechanisms to endow certain systems with homeostasis, for example the Unix operating system [4] and robotics [5]. Within a biological context, it is generally accepted that organism homeostasis is an emergent property of the interactions between the immune, neural and endocrine system. Taking this view, work in [6] discussed mechanisms inspired by the neural and endocrine systems and how these might be exploited in the context of robotic systems. However, there is a great deal of complexity issues when one examines the interactions of these three systems, therefore we have decided to focus on a single subsystem, the immune system, in an attempt to get a handle on the inherent complexity. In practice, and as it will be seen in this paper, it is almost impossible to draw lines between each of these systems, in particular the immune and endocrine systems, as there are so many types of interactions at so many different levels. In this position paper we examine the issues involved in creating a general extensible architecture for homeostasis for use in electronic systems that will endow homeostatic properties on engineered system. However, this is not a simple task and this position paper serves as a discussion on the issues regarding designing such an architecture.

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 216–227, 2007.

2 Biological Homeostasis and Biological Homeostatic Control Systems

The processes that encompass homeostasis are best understood by looking to the original definition [1]:

> The coordinated physiological processes which maintain most of the steady states in the organism are so complex and so peculiar to living beings involving, as they may, the brain and nerves, the heart, lungs, kidneys and spleen, all working cooperatively that I have suggested a special designation for these states, *homeostasis*. The word does not imply something set and immobile, a stagnation. It means a condition a condition which may vary, but which is relatively constant.

A present day reference on homeostasis, Vander's Human Physiology [2] acknowledges the stability provided by homeostasis is due to interactions of the immune, neural and endocrine systems; and that homeostasis also occurs individually within each one of these systems. Vander [2] opens with a chapter on homeostasis to provide a context to the whole book, the chapter describes homeostasis in terms of *homeostatic variables* and *set points* of those variables. Examples of homeostatic variables in the human body are blood glucose levels, or body temperature. The set points are the steady states (not necessarily equilibria) at which the system attempts to maintain these variables. Vander notes that over a given time period there may be massive variability in homeostatic variables, there is a rise in blood glucose after a meal, for example. But, if a time-averaged mean across that time period is taken and compared with consecutive time periods the behaviour is far more predictable. The control of the of homeostatic variables and set points is performed by *homeostatic control systems*. They are predominantly feedback systems, most often negative feedback, but positive feedback does also occur. Vander [2] supplies a list of general properties homeostatic control systems, which is reproduced here.

1. Stability of an internal environmental variable is achieved by balancing inputs and outputs. It is is not the absolute magnitudes of the inputs and outputs that matter, but the balance between them.
2. In negative feedback systems, a change in the variable being regulated brings about responses that tend to move the variable in the direction opposite the original change — that is, back toward the initial value (set point).
3. Homeostatic control systems cannot maintain complete constancy of any given feature of the internal environment. Therefore, any regulated variable will have a more-or-less narrow range of normal values depending on the external environmental conditions.
4. The set point of some variables regulated by homeostatic control systems can be reset — that is, physiologically raised or lowered.
5. It is not always possible for homeostatic control systems to maintain constancy in every variable in response to an environmental challenge. There is a hierarchy of importance, so that the constancy of certain variables may be altered markedly to maintain others at relatively constant levels.

Vander [2] discusses a number of other issues and examples of homeostatic control systems in context of the human body, we abstract these to produce a further list of desirable properties of homeostatic control systems.

1. *Prediction.* Vander [2] determines this as feed-forward regulation. In response to an environmental change the homeostatic control system manipulates the internal environment in order to avoid a deviation from a set point before it has happened.
2. *Innate and Adaptive Response.* The homeostatic control system is built up of innate and adaptive *reflexes* which are used to bring homeostatic variables back to set points. The innate reflexes are involuntary, unpremeditated and unlearned, and are instigated in response to a particular stimulus, internal or external. As one would imagine, adaptive reflexes are learned to correct unforeseen deviations from set points. Vander also states that all reflexes, innate or adaptive, are subject to further learning.
3. *Acclimatisation.* Although encompassed by both adaptive responses and resetting of set points, it is an important enough property in its own right. It represents the ability for a set point to semi-permanently change in response to semi-permanent change in the environment. To aid explanation we take the analogy in [2] of a runner who is asked to run for 8 consecutive days in a hot room (a room hotter than the runner's normal environment). Details of the runner's sweating are recorded. By the 8th day the runner starts to sweat earlier and in far greater quantities than the 1st day, this allows to the runner to limit the deviation of the temperature homeostatic variable from its set point. The 'sweating' homeostatic set point has acclimatised to the new environment. When the runner returns to running in the original environment the set point will, over a number of days, acclimatise back to the original.

3 The Immune System for Homeostasis

First, it is worth noting why we are attempting to construct artificial homeostatic systems using solely the immune system, apparently ignoring the neural and endocrine systems. All three systems are necessary for human homeostasis and none of the systems are singularly sufficient. When one investigates the immune system, it is clear that the endocrine system is inextricably linked to the immune system. Immune cytokine networks share many of the same functional properties of the endocrine system, and are in effect considered part of the endocrine system: therefore we are not ignoring the role of the endocrine system. There is clear evidence to suggest immune, neural and endocrine interactions [3], however, as previously mentioned we have excluded the neural system from our studies, as by doing so reduces the level of complexity that we are dealing with and allows us to focus our efforts at exploring the role of the immune system in body maintenance, a view held by some theoretical immunologists [7] and [9].

3.1 Cohen's Immune System

Cohen believes the role of the immune system is to repair and maintain the body. As the removal of pathogen is beneficial to the health of the body, defence against pathogen can be seen as just a special case of body maintenance. In order to achieve body maintenance, the immune system must select and regulate the inflammatory response according to the current condition of the body. This condition is assessed through the Co-respondence of both the adaptive and innate immune agents, which are required to recognise both the presence of pathogen (non-self antigen) and the state of the bodys own tissues (self antigen). The specificity of the immune response, therefore, is not just the discrimination of danger [11] or the distinction of self/non-self, but the diagnosis of varied situations, and the evocation of a suitable response. Degeneracy is a concept central to Cohen's ideas and is discussed in terms of immune receptors and cytokine networks. Degeneracy is defined as [20]: "The ability of elements that are structurally different to perform the same function of yield the same output." Cohens maintenance role of the immune system requires it to provide three properties: Recognition: to determine what is right and wrong, Cognition: to interpret the input signals, evaluate them, and make decisions. Action: to carry out the decisions.

3.2 Grossman's Tunable Responses

Grossman [8] sees the immune system as a system which reacts to perturbations, to changes in the environment rather than the specificity of any particular pathogen. Grossman's view is constructed around immune cells having with tunable activation thresholds, which control proliferation, differentiation and choice of effector function. The activation thresholds are tuned to a cell's recent excitation history (its interactions and interaction affinity) [8], a change in the environment will cause a change in the cells excitation. The rate and size of perturbation with respect to the cell's history is what ultimately determines the response of the cell. Grossman believes control of immune response (e.g. severity of attack, tolerance and memory) emerges out of the dynamics of a population of tunable cells [9]. Grossman provides a simple mathematical model for the tuning of cells in [8] and outlines potential biological evidence for the tuning in [10].

3.3 Appropriateness of Immune Inspiration

Cohen's and Grossman's theories concern the immune system as a whole, their arguments relate to interactions producing behaviour rather than analysis of immune machinery. They view the immune system in terms of maintenance and tolerance rather than attack and defence of invading pathogens. The concepts of Cohen and Grossman have commonalities with the concepts of biological homeostasis and homeostatic control systems, section 2. They provide immune theories that would seem to provide excellent inspiration for construction of a homeostatic control system. There are certainly conflicts between the two models of the immune system, Grossman requires some immune receptor specificity [8], and Cohen prefers to do away with specificity altogether [12]. However there is definitely common ground and opportunity to combine the two, in fact Cohen does precisely this in [12] to produce a model to describe T Cell behaviour.

4 An Architecture for Artificial Homeostasis

We will now discuss some of the issues that arise in attempting to construct an architecture for homeostasis based the notion of homeostatic control systems provided. Again, it should be noted that this is a position paper, we discuss potential problems and propose some tentative solutions.

It is our intention that if this architecture can be developed then systems built adhering to the rules of the architecture will then have the properties of a homeostatic system. Consequently the system should have an innate level of homeostasis and then adapt and acclimatise to the environment it is placed within. We now define some terms within our system, and say that the system is comprised of:

1. *Sensors.* These can sense from the environment.
2. *Homeostatic Variables and Set Points.* For the system to maintain homeostasis we must define what it means for the system to be homeostatic. The intention is that homeostatic variables are evaluated by functions on the sensors, and other internal variables of the system. Each of these are associated with a priority, to represent the importance of certain variables over others as mentioned in section 2.
3. *Actuators.* These can act to manipulate the environment.
4. *Homeostatic Responses.* Similar to the homeostatic variables and set points, the system requires innate methods to correct deviations from homeostatic set points. The responses would make use of the systems actuators.
5. *Tasks.* These describe the behaviour of our system.
6. *Homeostatic Control System.* This maintains the homeostatic variables at their set points, while allow the system to complete its tasks.

4.1 Splitting the Problem: Breaking into a Homeostatic System

Imagine that we have constructed a homeostatic control system, which is able to maintain homeostasis given the Sensors; Actuators; Homeostatic Variables, Set Points and Responses; and Tasks. The problem now is: can we sensibly and tractably split up a system into these components? Although choice of sensors, actuators and tasks are ultimately specific to problem domain of the system, there are still general considerations. It is important to understand the purpose of the system, this may seem obvious, but it raise some interesting issues. For example: is it more important for the system to complete its task, or is survival of the system (it not becoming irreparably damaged) more important? Is there only a single system performing a given task, or are there many systems? Therefore, is losing one or two systems an acceptable cost in order to complete the task? If this is the case the homeostatic control system could be a little more cavalier with the choices of homeostatic response, this should be reflected in innate definitions of the system.

Similarly there are some general points on choice of innate homeostatic variables, set points and responses. An interesting way to split the problem is to observe how the problem is split in biological systems. There is a natural hierarchy to biological systems, the homeostasis of an individual is maintained by the immune, neural and endocrine systems. The homeostasis of these systems is maintained by systems internal to them, and then the homeostasis of those systems is maintained by systems internal to them, and so on.

The homeostasis of artificial systems can be broken up in a similar manner. At the population; at the individual; at the tasks; the physical components of a system. Problem splits can be both logical and physical, clearly splitting by tasks represents a logical split whereas by components represents a physical split. There is no constraint on splits being entirely logical or entirely physical, one can imagine a system split both logically and physically with the split problem represented as a graph. Nodes in the graph would represent *homeostatic units*, and edges of the graph represent a communication of homeostasis information between homeostatic units of the system. This raises the questions: what information should be shared between homeostatic units and how should this information be shared? It is reasonable to envisage homeostatic variable deviation information propagating through this graph, but sharing of homeostatic responses and correlations is less obvious. Imagine a 4-wheeled robot with a homeostatic architecture split so that each of the four is controlled by four homeostatic units. Each wheel is mechanically identical and has identical sensors. One of the four discovers a correlation or response that is useful in predicting and avoiding flat tyres, how can this information be propagated to the other three wheels? The difficulty arises that each homeostatic unit is by intention self-organising; there is a black box element to these homeostatic units. We don't not know what coding scheme has be adopted in each of the four homeostatic units, to communicate the information we would have to isolate the information that deals with new discovery along with the other information it depends upon. Then a mechanism would be required to translate from the coding scheme of the discovering unit to the coding scheme of the other three units. We should note that we are not suggesting these steps be literally implemented, but that they hopefully can emerge as part of an appropriately immune inspired algorithm.

We return, briefly, to discussing logically splitting the problem and suggest a temporal heuristic may be useful. Systems are required to operate over a variety of time scales, it is a property of general systems including computational, psychological and social [17]. For example, parts of our system may need to respond on time scale t_1 and other parts may need to respond on timescale t_2. The difference between t_1 and t_2 may be so pronounced that from the point of view of t_1 operations on timescale t_2 occur instantly and atomically, and from the point of view of t_2, t_1 looks constant. This property is noted in Burns et al. [17] who take inspiration from psychology and real time systems to develop a formal framework of *timebands*. It can be used to describe and prove relationships between entities interacting at different timescales. No proof or analysis of interactions between timescales is needed here, in fact such analysis flies in the face of the purpose of our architecture. However, these time scales still provide a natural way to break up the system into homeostatic variables with timescales in common sharing hierarchical levels. It is worth noting that this hierarchy does not necessarily impose a order of importance, or that lower levels in the hierarchy must be completed before higher levels. It is just an intuition with which to split the problem and to characterise interactions between homeostatic variables and responses.

There is an issue concerning the granularity with which homeostatic set points are defined. The intention is to allow the architecture to discover an emergent method of maintaining homeostasis, as a consequence we do not want the set points to be defined with too fine a granularity. For example, homeostatic variables could be represented by a real numbers and if the set points are set at specific real target values, then the homeostasis of the system is very prescriptive. The solution to maintaining homeostasis reduces to an error minimisation problem, and biologically inspired techniques, although still useful, are no longer necessary. However, this overlooks the fact that it may be very hard to assign fixed target values to the set points to achieve the desired behaviour. Moreover, systems where this is a straight forward task are not the systems we are interested in bestowing with homeostatic properties. Set points should be defined in a fuzzier sense: intervals, minimisations, maximisations etc. For example a component in the system may have an operational temperature range in $[-10, 50]$ degrees Celsius, this range is an obvious choice for a set point.

A final point of discussion in the choice and the assignment of homeostatic variables is their priorities. In section 2 it is noted that there is a hierarchy of importance of homeostatic variables and the homeostasis of some variables may be sacrificed to maintain the homeostasis of others. Homeostatic variable priorities are a natural way to represent this property. The assignment of priorities to homeostatic variables will necessarily constrain the behaviour of the system. An incorrect intuition about priorities may lead to system failure in the worst case and unnecessarily restrictive behaviour in the best case. We suggest that priorities are assigned very carefully and potentially sparingly, perhaps assigning many homeostatic variables the same priority. The concept of the homeostatic control system sacrificing certain homeostatic variables in favour of others can still exist in a system with many homeostatic variables sharing the same priority. Many homeostatic variables can be allowed to deviate slightly from their set points to avoid a large deviation on a separate homeostatic variable.

Choosing innate responses to correct deviations from homeostatic set points is subject to similar issues as that of choosing homeostatic variables. We would like the homeostatic control system to discover good choices for homeostatic responses, so we do not necessarily need sophisticated innate responses, the hope being the homeostatic control system will discover them. A simple heuristic for choice, then, is all that is needed. We suggest a greedy response: each homeostatic variable is associated with the response that will best correct that variable, if perturbed, regardless of effect of that response on the rest of the system. For example in an autonomous robot system, the temperature homeostatic variable may be associated with a response to turn on the fan (pull the variable back towards its set point) and to turn off all motors (eliminate a potential source of the perturbation).

4.2 Homeostatic Control System

Before discussing how an immune inspired homeostatic control system may be constructed, it is useful to examine the issues that arise in attempting to maintain homeostasis on our new definition of a system. Immune inspiration or not our system is beginning to fit into Cohen's [7] three stage system: *Recognition*, sensors sense the environment which position the homeostatic variables. *Cognition*, based on the sensor values and homeostatic variables in relation to their set points we must decide on homeostatic responses to take. *Action*, the homeostatic responses act to move homeostatic variables back towards set points.

It is worth noting that linking a homeostatic response to the correction of a homeostatic variable is perhaps sufficient for an innate response, but it is definitely not sufficient for adaptive responses. To ease discussion we will define a new term: *homeostatic error* for a homeostatic variable, this is simply the distance (by whatever distance metric we care to choose) of a homeostatic variable from its set point. To reiterate, now with the new parlance, it is not sufficient to link adaptive homeostatic responses to homeostatic error. The homeostatic error is only a context with which to understand the sensor values.

We return to the properties of homeostatic systems given in section 2, and determine what our system must do given our current definitions:

1. *Arbitration* — given homeostatic variables with homeostatic error and the system's current understanding of the homeostatic responses, the control system must arbitrate between the possible responses to best maintain homeostasis.
2. *Correlation of Sensor and Homeostatic Error* — The system must correlate the sensor conditions under which homeostatic errors tend to arise.
3. *Response Learning* — The system must improve on innately supplied homeostatic response choices, by learning responses for the correlations learned in the previous step. This includes recovery from the failure of responses, if an actuator used in a response fails the system must use a different combination of actuations to achieve the desired response.
4. *Prediction* — Using the correlations and responses learned the system should predict the movement of homeostatic variables to avoid homeostatic error wherever possible.
5. *Acclimatisation* — This represents the systems ability to change its correlations and learned responses as the environment changes. This highlights the issue of *overspecificity*, the system becoming too specific to a given environment and failing when the environment changes. We want our environments change and so the system must be able to acclimatise to any new environment that arises.

An interesting way to think of the acclimatisation and the avoidance of overspecificity is in terms of robustness. Specifically robustness to changes in the environment, this can be examined in information theoretic terms using the mutual information [18], I, between the system S and the environment E:

$$I(S:E) = H(S) - H(S|E) \tag{1}$$

$H(S)$ is the entropy of the system and $H(S|E)$ is the conditional entropy of the system with respect to the environment. This conditional entropy term can be thought of as the amount information that is in the system that is not correlated to anything in the environment [18]. This represents an information excess, it allows the system to change neutrally (a change that does not effect the mutual information) without changing the systems behaviour. By a similar argument the environment can change neutrally, without the system noticing, if E changes in a way not correlated in S. The system and its behaviour can be robust to both internal and external changes. The problem now becomes one of striking the appropriate level of mutual information, to allow for acclimatisation but not overspecificity. This can be answered by examining the rate of change of the environment with the rate of change of the mutual information, that is the ability of the system to keep up with a changing environment. Insight is given observing by the specificity of the system to the environment at a given time, if the system too specific (overspecificity, or overtraining) a change in the environment will seem large from the point of view of the system, and so a large rate of change and the system may not be able to keep up. If the system is unspecific for the environment a change in the environment will seem small, there will be a small rate of change. However if the system is too unspecific changes in the environment will not cause appropriate acclimatisation and incorrect system behaviour.

Artificial systems have considerations of homeostatic control beyond the those that are obvious from biology. Sensors can fail and their ability to sense the environment to a specified range and resolution can degrade. The consequence of sensor degradation could cause an incorrect calculation of homeostatic error which in turn could cause an incorrect homeostatic response. Catastrophe can occur, a failure of the system to fulfil its purpose, in our terms: either failure of system survival or failure of the system to complete its task. Imagine a situation where the homeostatic response caused by an incorrect error perpetuates the true homeostatic error, which causes positive feedback and dramatic increases of homeostatic error to the point of catastrophe. Sensor failure is a far more malicious failure than response failure. Although it is impossible to completely eliminate the possibility catastrophe, with appropriate system design it is possible to reduce the probability of a catastrophic collapse. There is a natural physics provided by the environment, for example in a robot, it may be possible to estimate robot temperature as a function of motor speeds rather than using a thermometer. This highlights a need for multiple methods of evaluating the same homeostatic variable, this is an extension of an already identified desirable property of our system. Currently our system must correlate sensor data and homeostatic errors to better identify when homeostatic errors occur and when they are likely to occur. Now, we would like our system to correlate sensor data and homeostatic error in order to produce multiple methods of evaluating homeostatic variables. This is providing degenerate methods of evaluating homeostatic variables, this turns out to be exactly what we want.

Degeneracy and Redundancy. We have defined degeneracy in section 3.1. In contrast *redundancy* is characterised by multiple identical structures performing the same function [20]. A further important distinction is that although under certain conditions degenerate structures can perform the same function, under different conditions they can perform different functions. Degeneracy and redundancy are analysed in information theoretic terms in [19], the formulation is in terms of mutual information between subsets of inputs and outputs. They establish that [19]:

> A degenerate system, unlike a fully redundant one, is thus extremely adaptable to unpredictable changes in circumstances.

They stress that a degenerate system must have a degree of functional redundancy,there is scale between a fully redundant systems (everything performs the same function) and fully independent systems (a one to one mapping, everything performs different functions), degeneracy lies in the middle. Tononi et al. [19] also comment that degeneracy and complexity go hand in hand, and systems with high degeneracy have the potential for high complexity. Appropriate degenerate structures would seem to provide precisely the robustness for the balance between specificity and acclimatisation discussed earlier. Consequently degeneracy is a property that we would like at many levels of our systems. It makes a case for our systems being sensor rich (mimicking biology), and not just sensor redundant, to allow for sensor noise, degradation and failure; and to be able to represent the complex relationships between degenerate sensors.

Returning to the more general topic of our homeostatic control system; biological systems develop their homeostatic control through evolution species over many generations. Similar ideas are possible artificially if there have a population of identical systems maintaining homeostasis and sharing information on homeostasis. But, in the case of a single system how is performance evaluated? Imagine all homeostatic variables are at or within homeostatic set points, but a more efficient and preferred system operation exists and to get to this operation requires movement into homeostatic error. Is this movement ever possible? The movement clearly requires a homeostatic variable or task linked with a concept of efficiency, however it is likely that this will have a lower priority than other homeostatic variables. A lower priority than the homeostatic variables that are required to briefly delve into error to allow the efficiency error to drop. We do not provide an answer here, more a comment from Cohen [7] ”what works, works.”, if the total homeostatic error is at zero the system is successful, if the behaviour is not the desired one then the homeostatic variable definitions must be changed.

5 Towards an Immune Inspired Solution

There are themes that arise in the discussion of artificial homeostasis which clearly link to the immunological theories. The co-respondance of [7] and [9] combined with the tuning of [8] and the cytokine networks of [14], to provide the arbitration. The degeneracy of [7] providing the robustness through appropriate mutual information between the environment and the system. The population dynamics of [7] and [10] giving rise to necessary decisions, proliferation and memory.

Although there are many conflicting theories on the immune systems operation and function, there is less argument to the immune system's function on a basic mechanical level. We demonstrate analogies between the function of components of our homeostatic system and the function of some immune components. To begin with, there is analogy between the lymphatic system, and our homeostatic unit graph, the units and the lymph nodes being distributed locations in which the problem and the solution is determined. Clearly both systems have an innate/adaptive divide. At the innate level there are analogies between antigen presenting cells (APCs), macrophages, the context they present and the the innate variable evaluations and response. At the adaptive level there is analogy between the T Cells, B Cells and there proliferation and differentiation decisions and the adaptive recognition and response of the homeostatic system. These analogies serve little purpose in the design of an immune inspired solution, but are used to demonstrate that inspiration is needed from the whole immune process, not one individual part or concept.

5.1 Moving from Inspiration to Algorithms

Much of our inspiration is born from the arguments made Cohen [7] and Grossman [8], however a number of the arguments are neither scientifically verified nor their workings fully understood. We suggest taking a principled approach in the form of the conceptual framework [13], which may allow us to build immune algorithms which are correctly biologically grounded, better theoretically understood and ultimately more successful. The first step is to model mathematically and computationally the biology as it is explained by [7] and [8], to gain a better understanding of the biological processes and interactions involved. Distance along this road has already been travelled with modelling of degeneracy in both biological and artificial immune systems [19], [15] and [16]; with mathematical modelling of cytokine networks [14]; and with model combining some of the out of Cohen and Grossman [12]. This leaves much of the work in understand to theories put forward by Grossman: [8], [9], [10], which we believe has the potential to be some of the most fruitful immune inspiration we have outlined.

6 Conclusions

We feel that it is indeed possible to build an immune inspired architecture for homeostasis. The greatest chances of success lie through: proper understand of the homeostasis problem; appropriate choice of immunological inspiration; and considered understanding of the biology behind the inspiration.

References

1. Cannon, W.B.: The Wisdom of the Body. Norton, New York (1932)
2. Widmaier, E.P., Raff, H., Strang, K.T.: Vander's Human Physiology: The Mechanisms of Body Function, 10th edn. Mc-Graw Hill, New York (2006)
3. Besedovsky, H.O., Del Rey, A.: Introduction: immune-neuroendocrine network. Front. Horm. Res. 29, 1–14 (2002)
4. Somayaji, A., Forrest, S.: Automated Response Using System-Call Delays. In: 9th USENIX Security Symposium (2000)
5. Brooks, R.: A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation 2(1), 14–23 (1986)
6. Neal, M., Timmis, J.: Timidity: A Useful Mechanism for Robot Control? Informatica 27(4), 197–204 (2003)
7. Cohen, I.R.: Tending Adams Garden: Evolving the Cognitive Immune Self. Elsevier Academic Press, Amsterdam (2000)
8. Grossman, Z., Paul, W.E.: Adaptive Cellular Interactions in the Immune System: The Tunable activation threshold and significane of subthreshold response. PNAS 89, 10365–10369 (1992)
9. Grossman, Z.: Cellular Tolerance as a Dynamic State of the Adaptable Lymphocyte. Immunological Reviews. No. 133 (1993)
10. Grossman, Z., Paul, W.E.: Autoreactivity, dynamic tuning and selectivity. Current Opinion in Immunology 13, 687–698 (2001)
11. Matzinger, P.: The Danger Model: A renewed sense of Self. Science 296, 301–305 (2002)
12. Hershberg, U., Solomon, S., Cohen, IR.: What is the basis of the immune system's specificity? Mathematical Modelling and Computing in Biology and Medicine, 377–384 (2003)
13. Stepney, S., Smith, R.E., Timmis, J., Tyrrell, A.M., Neal, M.J., Hone, A.: Conceptual Frameworks for Artificial Immune Systems. Int. J. Unconventional Computing 1(3), 315–338 (2005)
14. Hone, A., van den Berg, H.: Modelling a Cytokine Network. In: FOCI 2007 (2007)
15. Andrews, P.S., Timmis, J.: A computational model of degeneracy in a lymph node. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2005. LNCS, vol. 4163, pp. 164–177. Springer, Heidelberg (2005)
16. Mendao, M., Timmis, J., Andrews, P.S., Davies, M.: The Immune System in Pieces: Computational Lessons from Degeneracy in the Immune System. In: FOCI 2007 (2007)
17. Burns, A., Hayes, I.J., Baxter, G., Fidge, C.J.: Modelling Temporal Behaviour in Complex Socio-Technical Systems. Techinical Report YCS-2205-390. Department of Computer Science, University of York (2005)
18. Weeks, A., Stepney, S., Polack, F.A.C.: Neutral Emergence: a proposal. In: Symposium on Complex Systems Engineering, RAND Corporation (2007)
19. Tononi, G., Sporns, O., Edelman, G.M.: Measures of Degeneracy and Redundancy in Biological Networks. PNAS 96, 3257–3262 (1999)
20. Edelman, G.M., Gally, J.A.: Degeneracy and complexity in biological systems. PNAS 98(24), 13763–13768 (2001)

Modeling Migration, Compartmentalization and Exit of Naive T Cells in Lymph Nodes Without Chemotaxis

Johannes Textor[1] and Jürgen Westermann[2]

[1] Institut für Theoretische Informatik
[2] Institut für Anatomie
Universität zu Lübeck, 23538 Lübeck, Germany
textor@tcs.uni-luebeck.de

Abstract. The migration of lymphocytes through secondary lymphoid organs was believed to be mainly controlled by chemokine gradients. This theory has recently been called into question since naive lymphocytes observed *in vivo* by two-photon microscopy show no evidence of directed migration. We have constructed a simple mathematical model of naive T cell migration in lymph nodes that is solely based on local mechanisms. The model was validated against findings from histological analysis and experimentally determined lymphocyte recirculation kinetics. Our results suggest that T cell compartmentalization in lymph nodes can be explained without long-range chemokine gradients. However, the T cell residence time predicted by our model is significantly lower than observed *in vivo*, indicating the existence of a mechanism which alters the T cell random walk over time.

1 Introduction

Immune responses are mounted and supervised in secondary lymphoid organs (SLOs). These functions are crucial: loss of all secondary lymphoid organs leads to death. Among the several kinds of SLOs, i.e. lymph nodes, Peyer's patches and the spleen, the mass of lymph nodes makes up about 60% [1].

Unlike many other organs which consist mainly of *resting* cells, SLOs mostly contain highly *motile* cells: Lymphocytes perpetually enter, move within, and exit from SLOs. This restlessness is important because a previously unseen antigen is sometimes only recognizable by a few dozen out of several millions of lymphocytes. Since infections are usually local, the right cell being at the wrong place may mean that the immune response is mounted too late. Additionally, many essential functions of lymphocytes are *contextual*, i.e. restricted to a certain environment. For instance, somatic hypermutation of B cells is mainly restricted to germinal centers. Thus, in order to understand lymphocyte *function*, it is essential to study their *migration* [2].

On their endless journey, lymphocytes are often guided by chemotaxis [3,4,5]. For example, chemokines control the recruitment of circulating lymphocytes

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 228–239, 2007.

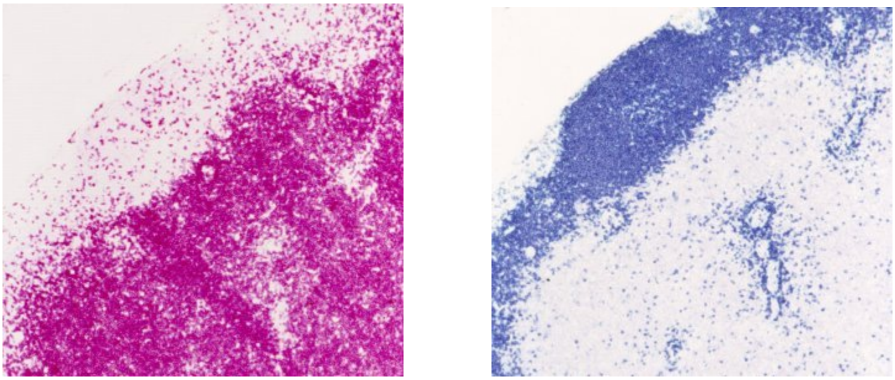


Fig. 1. Cortex and paracortex in slices of an uninfected lymph node of a rat (source: own work) with colored T cells (left) and colored B cells (right). High endothelial venules appear as “holes” in the paracortex.

into SLOs. Knowledge of these mechanisms has important clinical implications: chemokine receptors controlling lymphocyte migration have repeatedly proved to be excellent targets for the inhibition of inflammatory diseases like atopic dermatitis or multiple sclerosis [6]. Unsurprisingly, it has been proposed that chemokines also control the migration of lymphocytes *within* lymph nodes and other SLOs (e.g. [4]).

However, this theory has recently been called into question [7]: By two-photon microscopy, it has become possible to observe individual cells in the intact lymph node *in vivo* [8,9]. It was found that naive T cells show no sign of directed migration, as they would under the influence of chemokine gradients. Rather, their motion is best described as a random walk [7,10,11,12,13].

Our goal is a better understanding of these new results. In particular, we are interested in three aspects: (i) *migration* of naive T cells within lymph nodes; (ii) *compartmentalization* (see Fig. 1) of the constantly moving cells and; (iii) *exit* of naive T cells from the lymph node. We know that the entry of lymphocytes into SLOs is controlled by chemokine gradients – does the same apply to migration, compartmentalization and exit?

To address these questions, we have constructed a simple mathematical model that does not use chemotaxis, but is solely based on local mechanisms. Compartmentalization is explained by assuming that naive T cells *prefer* to walk along the fibers of the FDC network which makes up the paracortex, but *retain* their normal motility when outside of this network. Entry and exit are modelled as uniformly distributed processes in the paracortex and the medulla, respectively. We have found that these minimal assumptions give rise to dynamics that reasonably match recirculation dynamics and compartmentalization observed *in vivo*. Our results suggest that chemotaxis is not necessary to explain the forming of the lymph node compartments. However, the T cell residence time predicted by our model is significantly lower than in reality. This may indicate the existence of a mechanism which alters the motility of naive T cells over time, as it was shown to be the case for T cells which are stimulated by antigen [13].

2 Background

The basic structure of a lymph node is shown in Fig. 2. Both B and T cells enter through the *high endothelial venules* in the paracortical area. These capillary vessels bear molecules which "recruit" circulating lymphocytes from the blood stream. Presentation of processed antigen to T cells by dendritic cells takes place in the *paracortex*. In this densely packed environment, T cells and dendritic cells have frequent, short-lasting contacts. B cells, on the other hand, migrate to the *cortex* where they screen *follicular dendritic cells* for native antigen. The existence of designated lymph node compartments for B and T cells thus makes the search for antigen of both subpopulations more efficient. Upon activation, both B and T cells migrate towards the border between cortex and paracortex. Here, they exchange the costimulatory signals that initiate germinal center formation and with it the Th2 immune response [14]. This is why the existence of a real physical barrier between the two compartments is neither likely nor desirable: upon antigen challenge, it would stand in the way of germinal center formation.

Due to the absence of a physical barrier, it was long unclear how the separation between T and B cell areas in lymph nodes is established and maintained. Visually, the two areas are not distinguishable. Most likely, paracortex and cortex are defined by different types of stromal cells. In the paracortex, the *fibroblastic reticular cells* (FRCs) dominate. The best defined stromal cell subset in the cortex are the follicular dendritic cells (FDCs) that present native antigen to B cells [5,15]. Bajénoff et al. [10] have recently demonstrated *in vivo* that T cells move along the fibers of the FRC network while B cells walk along the FDC network.

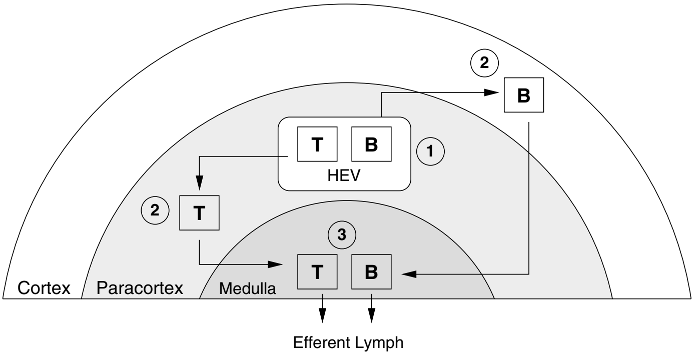


Fig. 2. Schematic representation of lymphocyte migration through the lymph node: (1) Lymphocytes enter the lymph node through high endothelial venules in the paracortex. (2) T cells remain in this compartment, while B cells migrate to the cortex. (3) B and T cells exit the lymph node via efferent lymphatic vessels which drain the medulla.

Still, these findings do not completely rule out the possibility that chemokines play a role in defining cortex and paracortex. First of all, the mentioned migration of activated T and B cells towards the cortex/paracortex boundary is controlled by chemokines made by the FRCs and FDCs [4]. These gradients might also affect naive lymphocytes. Due to technical issues, two-photon imaging experiments are limited to a timeframe of about 1-2 hours [8]. This is short compared to the 24 hours lymphocytes typically spend in lymph nodes. Weak chemokine gradients that only create a small bias in lymphocyte motility might not be detected by this method. Such gradients might have an important effect on the long-term dynamics, especially on the exit process. Formal modeling is helpful to shed light on these contradicting theories.

Our starting point was a simple hypothesis on how cortex and paracortex are formed: It makes no difference if T cells are released in the cortex or in the paracortex. They retain their normal motility pattern (this has been confirmed by Miller et al. [11]) and, when inside the cortex, do not use chemotaxis to find back to the paracortex. When approaching the cortex from the paracortex, the FRC network becomes more sparse. Under these circumstances, T cells will *preferentially*, but not exclusively, move along the remaining FRC fibers. This simple effect should already give rise to compartmentalization.

This gave rise to the idea of constructing a simple model which does not include chemotaxis, but is only based on local mechanism as the one described above, and validate it against experimental results from histological analysis and recirculation experiments. We hoped that this model would give us a better understanding of the nature of the boundary and compartments generated by the defined mechanism. Moreover, it would show whether the kinetics of lymphocyte recirculation could really be the result of a simple uniform random walk.

3 The Model

Several authors (e.g. [16,17]) proposed agent-based models of cellular interactions in the lymph node. These models are very useful to study the complex emergent aspects of the immune response like affinity maturation. However, they are too complex for our purpose, since we are focusing on naive cells. Beltman et al. [18,19] studied the dynamics of naive T cell migration using a cellular Potts model which focuses on the paracortex. This model is impressively consistent with experimental results from two-photon microscopy. Yet, we found it too complicated to incorporate medulla and paracortex into this model and decided to start with a simpler approach. Stekel et al. [20,21] studied the dynamics of lymphocyte recirculation using differential equations, but they did not model the lymph node compartments explicitly.

Compared to these approaches, our model is situated on an intermediate (mesoscopic) scale. We abstract from the individual cells and use scale-free concentrations instead, but represent the lymph node compartments explicitly. We have deduced the sizes and shapes of the compartments from the literature (see table 1) and from our own experiments and archived microtome slices. However,

Table 1. Parameters of our model with their respective meanings, default values and sources. Microtome sections from our own archives were used for estimating σ and e.

Symbol	*Space*	*Default*	*Meaning*	*Sources*
$r_{\mathcal{M}}$	$\mathbb{R}^+$	$300\mu m$	Base radii of medulla, cortex and paracortex	[10,15,25]
$r_{\mathcal{C}}$		$450\mu m$		
$r_{\mathcal{P}}$		$500\mu m$		
a_x	$[0,1]$	1	Relative sizes of ellipsoid main axes	[15,26]
a_y	$[0,1]$	.8		
a_z	$[0,1]$	.5		
$\boldsymbol{\sigma}$	$[0,1]$	0.88	"Stickiness" of T cells to paracortex, related to T cell concentrations by 7	Cell counting on microtome sections
$\boldsymbol{e}(t,v)$	$[0,1]$	$1/\lvert\mathcal{P}\rvert$	Spatiotemporal distribution of T cell entry in paracortex	n/a
$\boldsymbol{m}$	$[0,1]$	$67\frac{\Delta s^2\,\mathrm{min}}{\Delta t\,\mu m^2}$	T cell random walk motility	[11,12,7]
$\boldsymbol{l}$	$[0,1]$	0.005	Speed of exit process in medulla	Visual matching to microtome sections
Δs	$\mathbb{R}^+$	$20\mu m$	Spatial grid resolution, corresponds to mean length of FRC/FDC fibers	[10]
Δt	$\mathbb{R}^+$	1min.	Temporal resolution	

it must be taken into account that lymph node morphology varies considerably. For example, many lymph nodes show a concave boundary from medulla to paracortex (see e.g. [15]) instead of the convex one used in our model.

We focus on T cells since the available data on their motility patterns is much more detailed and reliable than that of B cells. This is partially due to technical difficulties with unambiguously monitoring B cells in the paracortex [10]. We plan to gradually incorporate B cells into our model as more reliable experimental results become available.

Our model is based on a three-dimensional grid. The edges of this grid represent the network of FRC and FDC processes in the lymph node. As shown in Fig. 3, the grid is subdivided into three zones, which represent the cortex, the paracortex and the medulla. The T cells are not represented individually, but as abstract, scale-free concentrations, which allows for larger-scale simulations, but sacrifices stochasticity. Our assumption that T cells migrate only by uniform random motion, except on the border from paracortex to cortex where they are more likely to remain in the paracortex, then corresponds to a diffusion process of the T cell concentration.

In the next two sections, we give a detailed formal definition of the model. It is primarily aimed at those who wish to reproduce our results and is thus written in a rather condensed form. Readers who are unfamiliar with lattice diffusion models should consult additional literature. For example, a similar two-dimensional model has been used to simulate the spread of Influenza infections on a cell monolayer [22]. A general introduction to models of diffusion which also covers reaction-diffusion and chemotaxis is given in Murray [23], Chap. 11.

3.1 Basic Structure and Evolution Equations

Let $\mathcal{LN} \subset R^3$ be a simply-connected smoothly bounded region (e.g. an ellipsoid) that is subdivided into three pairwise disjoint regions $\mathcal{M}, \mathcal{P}, \mathcal{C}$. We discretize $\mathcal{LN}$ by an isotropic cubic lattice $L = (V, E)$ with vertices $V = \{(n_x \varDelta s, n_y \varDelta s, n_z \varDelta s) : n_x, n_y, n_z \in \mathbb{Z}\}$ and edges $E = \{(v, v') \in V \times V : ||v - v'|| = 1\}$. The set of neighbours of $v \in \mathcal{LN}$ is written as $v^+(v) = \{w \in V : (v, w) \in E \wedge w \in \mathcal{LN}\}$. We additionally define $c^+(v) = v^+(v) \cap \mathcal{C}$. The interval $\{x \in \mathbb{R} : 0 \leq x \leq 1\}$ is denoted by $[0, 1]$.

An edge coloring $f(v, w) : E \mapsto \{0, 1\}$ represents the network of stromal follicular dendritic cells in the paracortex and fibroblastic reticular cells in the cortex: $f(v, w) = 1 \Leftrightarrow v \in \mathcal{C} \vee w \in \mathcal{C}$. The boundary between the compartments is denoted by $\mathcal{B} = \{v \in \mathcal{P} : c^+(v) \neq \emptyset\}$.

The distribution of T cells across the lymph node at time t is given by the function $T : \mathbb{R} \times \mathbb{V} \mapsto \mathbb{R}_0^+$. Our model evolves in discrete time steps $n\varDelta t, n \in \mathbb{N}$. Entry, random walk, and exit of T cells are modeled by evolution equations, analogous to finite difference approximations of differential equations.

Three kinetic parameters apply during each time step t: $\boldsymbol{e}(t, v) \in \mathbb{R}_0^+$ denotes the unitless amount of new T cells that enter a vertex $v \in \mathcal{P}$ from HEVs, which are not modeled explicitly. From every $v \in V$, a fraction $\boldsymbol{m} \in [0, 1]$ out of $T(t, v)$ moves to one of the neighbouring vertices. For $v \in \mathcal{M}$, a fraction $\boldsymbol{l} \in [0, 1]$ of $(1 - \boldsymbol{m})T(t, v)$ exits (leaves) from v. The structural parameter $\boldsymbol{\sigma} \in [0, 1]$ is a measure of the "stickiness" of T cells to FRC fibers. Altogether, the evolution equation of $T(t, v)$ for $v \in \mathcal{P}$ is defined as follows:

$$\begin{aligned} T(t + \varDelta t, v) = \boldsymbol{e}(t, v) + (1 - \boldsymbol{m})\, T(t, v) + \sum_{w \in v^+(v) \setminus \mathcal{B}} \frac{\boldsymbol{m}}{|v^+(w)|} T(t, w) \\ + \sum_{w \in v^+(v) \cap \mathcal{B}} \frac{|v^+(w)| - (1 - \boldsymbol{\sigma})|c^+(w)|}{|v^+(w) \setminus c^+(w)|} \frac{\boldsymbol{m}}{|v^+(w)|} T(t, w) \end{aligned} \tag{1}$$

On $v \in \mathcal{M}$, we use:

$$\begin{aligned} T(t + \varDelta t, v) = \boldsymbol{l}\, (1 - \boldsymbol{m})\, T(t, v) + \sum_{w \in v^+(v) \setminus \mathcal{B}} \frac{\boldsymbol{m}}{|v^+(w)|} T(t, w) \\ + \sum_{w \in v^+(v) \cap \mathcal{B}} \frac{|v^+(w)| - (1 - \boldsymbol{\sigma})|c^+(w)|}{|v^+(w) \setminus c^+(w)|} \frac{\boldsymbol{m}}{|v^+(w)|} T(t, w) \end{aligned} \tag{2}$$

And for $v \in \mathcal{C}$, we obtain:

$$\begin{aligned} T(t+\Delta t, v) = (1-\boldsymbol{m})\, T(t,v) + \sum_{w \in v^+(v) \setminus \mathcal{B}} \frac{\boldsymbol{m}}{|v^+(w)|} T(t,w) \\ + \sum_{w \in v^+(v) \cap \mathcal{B}} (1-\boldsymbol{\sigma}) \frac{\boldsymbol{m}}{|v^+(w)|} T(t,w) \end{aligned} \tag{3}$$

Setting $\boldsymbol{e}(t,v) = \boldsymbol{l} = 0$ for all v and t, (1) to (3) reduce compartment-wise to a discrete version of the continuous diffusion equation

$$\frac{\partial T}{\partial t} = D \nabla^2 T \qquad \text{where } D \frac{\Delta s^2}{\Delta t} = \boldsymbol{m}$$

which describes a discrete random walk. On the boundary of $\mathcal{LN}$, the random walk is "myopic": Since the fraction of T that is transferred to the neighbouring nodes is always $\boldsymbol{m}$, a higher relative amount is transferred along each adjacent edge. Thus, in equilibrium, boundary vertices v' hold a fraction of $v^+(v')/6$ of the concentration of non-boundary cells. This must be taken into account when calculating region-wise mean concentrations.

From numerical analysis, it is known that a finite difference approximation of the diffusion equation where $D\Delta t/\Delta s^2 > 0.5$ is unstable. In our model, $\boldsymbol{m} > 0.5$ could lead to a periodic random walk ("chessboard pattern") and thus to meaningless results. Thus, for explicit simulations, Δt must be sufficiently small.

The lattice spacing Δs may be set to the mean stromal cell network spacing of $20\mu m$ [10], but other sufficiently small values are also valid as long as $\boldsymbol{m}$ is scaled to reproduce the motility coefficient of T cells of about $67\mu m^2/\text{min}$ [11,12]. Reasonable values for $\boldsymbol{l}$ and $\boldsymbol{\sigma}$ will be derived in Sec. 4.

3.2 Lattice Topology

Size, shape and structure of $\mathcal{LN}$ are characterized by the 6 parameters $r_\mathcal{M} < r_\mathcal{P} < r_\mathcal{C} \in \mathbb{R}_0^+, a_x, a_y, a_z \in [0,1]$. $\mathcal{M}, \mathcal{P}$ and $\mathcal{C}$ are each delimited by regular ellipsoids as follows:

$$\mathcal{M} = \left\{ (x,y,z) \in \mathbb{R}^3 : \left(\frac{x^2}{a_x^2} + \frac{y^2}{a_y^2} + \frac{z^2}{a_z^2} \right) \le r_\mathcal{M} \right\} \tag{4}$$

$$\mathcal{P} = \left\{ (x,y,z) \in \mathbb{R}^3 : \left(\frac{x^2}{a_x^2} + \frac{y^2}{a_y^2} + \frac{(z - r_\mathcal{P} + r_\mathcal{M})^2}{a_z^2} \right) \le r_\mathcal{P} \right\} \setminus \mathcal{M} \tag{5}$$

$$\mathcal{C} = \left\{ (x,y,z) \in \mathbb{R}^3 : \left(\frac{x^2}{a_x^2} + \frac{y^2}{a_y^2} + \frac{(z - r_\mathcal{C} + r_\mathcal{M})^2}{a_z^2} \right) \le r_\mathcal{C} \right\} \setminus \mathcal{P} \tag{6}$$

This basic structure is illustrated in Fig. 3. Note that cortex and medulla are not connected, which is not always the case [15].

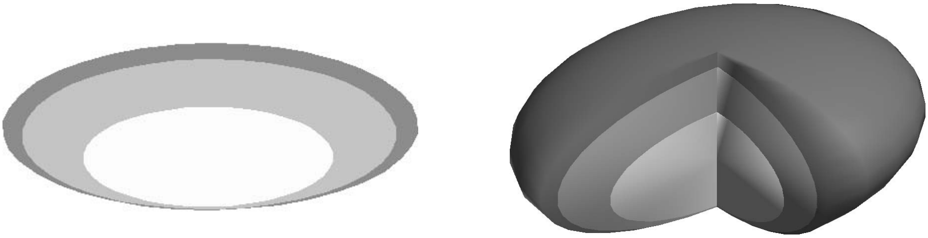


Fig. 3. Structure of our lymph node model. Medulla (innermost region), paracortex and cortex (outermost region) are delimited by ellipsoids as defined in Sec. 3.2. Left: 2D planar section. Right: 3D structure. Parameters are given in table 1.

3.3 Assumptions and Limitations

The model presented here is still extremely simplified. The standard random walk is not a realistic model of cell migration, as pointed out by Beltman et al. [18]: "abrupt directional changes are not possible because it takes time to displace the complicated internal structure that brings about motion". Indeed, histograms of T cell turning angles generated from intravital microscopy data show a significant deviation from a true random walk [19,13]: smaller turning angles are favoured. Moreover, T cells move in rhythmic patterns where phases of high velocities alternate with pauses.

However, from statistics, we know that most processes that behave "roughly" like a random walk and have a finite mean square step size will spread out like normal diffusion. This follows from the central limit theorem [24]. Since we are focusing on the mesoscopic to macroscopic scale, approximating the motility of T cells by a standard random walk should suffice as long as the T cell motility coefficient [11] is reproduced.

The absence of B cells is problematic for the analysis of compartmentalization. Effects like expulsion of T cells from the cortex are not taken into account. However, assuming a standard random walk, the tracks of individual cells are independent. Thus, adding B cells would not change the resulting dynamics.

Finally, entry and exit are modelled in a too simple manner. High endothelial venules are not explicitly represented, new lymphocytes just "appear" in the lymph node. We accept this for now since HEVs only make up about 1% of the paracortical area [25]. The exit process has not been modelled more explicitly since the amount of time T cells spend in the medulla is rather small [6]. All these simplifications have to be taken into account when interpreting our results.

4 Results

The default parameters used in our simulations are given in table 1 along with the corresponding literature references. However, some of the parameters could not be inferred from literature. The choice of the entry term $e(t, v)$ depends on the distribution of high endothelial venules (HEVs) across the paracortex. Even though it is generally assumed that HEVs tend to lie close to the cortex, we are not aware of any reliable data on this. So we initially assume that the HEVs are uniformly distributed and set $\boldsymbol{e}(t,v) = 1/|\mathcal{P}|$. Values for $\boldsymbol{\sigma}$ and $\boldsymbol{l}$ will be derived in the next section.

4.1 Compartmentalization

In order to investigate the influence of the stickiness parameter $\boldsymbol{\sigma}$ on the concentrations of T cells in cortex and paracortex, we have calculated the equilibrium concentration $\bar{T}$ where for all $v \in V : \bar{T}(t+\Delta t, v) = \bar{T}(t,v)$. This can be done by combining 1 to 2 with the equilibrium condition into a system of linear equations with the unique solution $\bar{T}$. We have found that if $\boldsymbol{e}(t,v)$ is sufficiently small with respect to $\bar{T}(t,v)$ and the variance of $\bar{T}(t,v)$ is sufficiently small across $\mathcal{C}$, the paracortex/cortex ratio of T cell concentration is approximately

$$\frac{\sum_{v\in\mathcal{C}} \bar{T}(t,v)/v^+(v)}{\sum_{v\in\mathcal{P}} \bar{T}(t,v)/v^+(v)} \approx \frac{1}{|\mathcal{B}|} \sum_{v\in\mathcal{B}} \frac{|v^+(v)| - (1-\boldsymbol{\sigma})|c^+(v)|}{(1-\boldsymbol{\sigma})|v^+(v) \setminus c^+(v)|} \tag{7}$$

Note that 7 depends only on $\boldsymbol{\sigma}$ and, to a lesser extent, on the shape of $\mathcal{B}$. Generally, the structure of $\mathcal{LN}$ is not important. This makes sense biologically since compartmentalization is still observed in morphologically degenerate lymph nodes [26]. Based on the approximation, the paracortex/cortex concentration ratio of T cells of about 11 (estimation based on counting cells on 2D microtome sections) should be reproduced by setting $\boldsymbol{\sigma}$ to about 0.88. Using our default parameters, this is indeed the case (see Fig. 4). Based on this result, we estimated $\boldsymbol{l}$ by visually matching slices of $\bar{T}$ to 2D microtome sections from entire lymph nodes ([15], own archives). $\boldsymbol{l} = 0.995$ generated both a sufficiently homogeneous concentration in the paracortex and a reasonable decrease of concentration in the medulla.

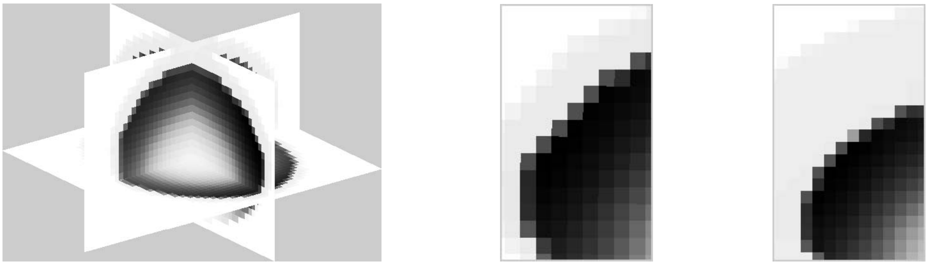


Fig. 4. Equilibrium concentration of 1 to 3 with default parameters (table 1). Left: 3D arrangement of xy,yz and xz slices. Center: Section similar to the one shown in Fig. 1, with sharply bounded compartments ($[\mathcal{P}]/[\mathcal{C}]$ ratio: 10.54; 7 gives 11.88). Right: 3-fold enlargement of the cortex ($r_{\mathcal{C}} = 0.6$mm) hardly influences the boundary ($[\mathcal{P}]/[\mathcal{C}]$ ratio: 10.62; 7 gives 11.98).

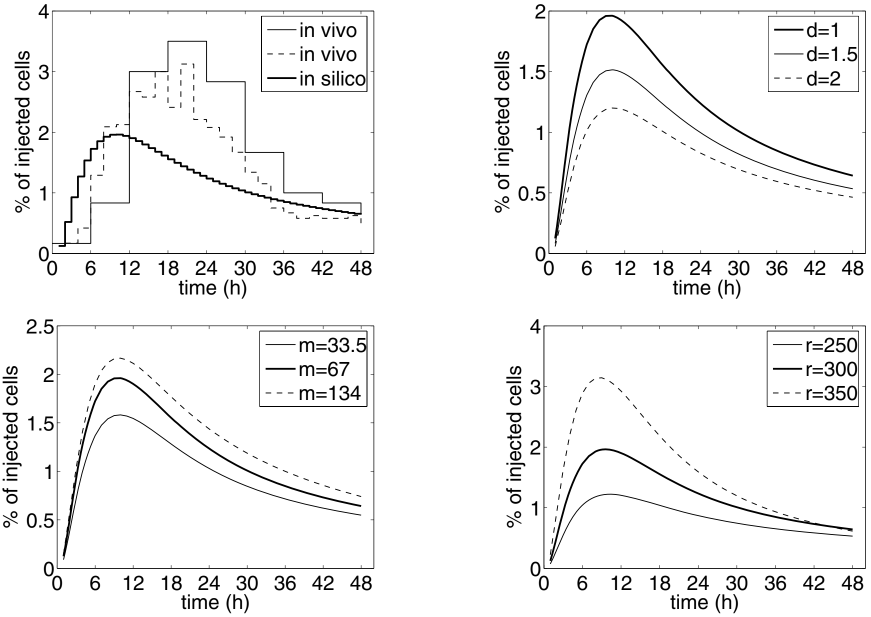


Fig. 5. Rates at which injected lymphocytes leave the lymph node plotted as a function of time. Top left: percentages of injected lymphocytes reappearing in thoracic duct (dashed: [27], collected in 2h-intervals; solid: [28], collected in 6h-intervals) plotted vs. the exit rate of our model with default parameters. Top right: The entire lymph node was scaled by d. Lymph node diameters between 1 and 2 mm are typical for mice, rats and humans; setting $d = 10$mm would reduce the peak to 0.2%. Bottom left: Modification of the T cell motility coefficient to 50% and 200% of its original value of $67\mu m^2$/min [11]. Bottom right: The kinetics are highly sensitive to modifications of the radius of the medulla (legend units are given in μm).

4.2 Recirculation Kinetics

Upon exit from lymph nodes, lymphocytes enter lymphatic vessels which drain into the thoracic duct. In a common experimental setup, labelled lymphocytes are injected into the blood. The reappearance of these lymphocytes in the thoracic duct is then measured as a function of time. The characteristics of these *recirculation kinetics* have been independently confirmed by many authors (see [20] for various sources) and are thus suitable for the validation of our model.

To be able to do this, we have included the process of lymphocyte recruitment in HEVs. Upon injection, the lymphocyte labelling index in the blood decreases exponentially. 50% of the initial value are reached after about 30 min [25,27]. After 2 hours, 50% of lymphocytes have migrated through the wall of the HEV into the paracortex [6]. We combined these two processes by setting $\boldsymbol{e}(v,t) = \exp(-\ln(2)(t)/150(t+1))$ (since $\Delta t = 1$min).

Results are shown in Fig. 5: Our results are comparable to those obtained *in vivo*, but the mean T cell residence time is significantly lower. There are several possible reasons for this: (i) a fraction of the injected lymphocytes have long-lasting contacts with DCs and thus remain longer in the lymph node, which is not accounted for in our model; (ii) T cell migration might be non-uniform in time, e.g. the cells start slowly upon lymph node entry, accelerating their movement after some time when no matching antigen is found; (iii) T cells could be retained in the lymph node by chemokines; however, this is unlikely, since the S1P chemokine gradient which exists between blood and SLOs rather seems to guide T cells out of the lymph node [29]. Our estimation for now is that (ii) and (iii) are responsible for the difference in residence time, since T cell priming by dendritic cells has been shown to occur in three distinct phases with different motilities [13].

5 Conclusions and Future Work

We have created a simple mesoscopic-scale model of naive T cell entry, migration and exit in the lymph node. We have shown how a simple local mechanism may give rise to compartmentalization into cortex and paracortex. Explicit simulations of recirculation experiments have yielded results that are comparable to those found by experiments, but show a significantly lower T cell residence time. This indicates the existence of a mechanism which alters the T cell random walk over time.

The next steps will be to model the exit process more explicitly and to gradually incorporate B cells as more reliable results on their migratory patterns become available. But most importantly, a profound mathematical understanding of the proposed model has to be gained.

References

1. Westermann, J., Pabst, R.: Distribution of Lymphocyte Subsets and Natural Killer Cells in the Human Body. Clin. Investig. 70, 539–544 (1992)
2. Von Andrian, U.H., Mackay, C.R.: T-Cell Function and Migration. Two Sides of the Same Coin. N. Engl. J. Med. 343(14), 1020–1032 (2000)
3. Moser, B., Loetscher, P.: Lymphocyte Traffic Control by Chemokines. Nat. Immunol. 2(2), 123–128 (2001)
4. Reif, K., Ekland, E.H., Ohl, L., Nakano, H., Lipp, M., Förster, R., Cyster, J.G.: Balanced Responsiveness to Chemoattractants from Adjacent Zones Determines B-Cell Position. Nature 416, 94–99 (2002)
5. Cyster, J.G., Ansel, K.M., Reif, K., Ekland, E.H., Hyman, P., Tang, H.L., Luther, S.A., Ngo, V.N.: Follicular Stromal Cells and Lymphocyte Homing to Follicles. Immunol. Rev. 176, 181–193 (2000)
6. Westermann, J., Engelhardt, B., Hoffmann, J.: Migration of T Cells In Vivo: Molecular Mechanisms and Clinical Implications. Ann. Intern. Med. 135, 279–295 (2001)
7. Wei, S.H., Parker, I., Miller, M.J., Cahalan, M.D.: A Stochastic View of Lymphocyte Motility and Trafficking Within the Lymph Node. Immunol. Rev. 195, 136–159 (2003)
8. Cenk, S., Mempel, T.R., Mazo, I.B., Von Andrian, U.H.: Intravital Microscopy: Visualizing Immunity in Context. Immunity 21, 315–329 (2004)
9. Halin, C., Mora, J.R., Sumen, C., Von Andrian, U.H.: In Vivo Imaging of Lymphocyte Trafficking. Annu. Rev. Cell Dev. Biol. 21, 581–603 (2005)

10. Bajénoff, M., Egen, J.G., Koo, L.Y., Laugier, J.P., Brau, F., Glaichenhaus, N., Germain, R.N.: Stromal Cells Networks Regulate Entry, Migration, and Territoriality in Lymph Nodes. Immunity 25, 1–13 (2006)
11. Miller, M.J., Wei, S.H., Parker, I., Cahalan, M.D.: Two-Photon Imaging of Lymphocyte Motility and Antigen Response in Intact Lymph Node. Science 296, 1869–1873 (2002)
12. Miller, M.J., Wei, S.H., Cahalan, M.D., Parker, I.: Autonomous T Cell Trafficking Examined In Vivo with Intravital Two-Photon Microscopy. PNAS 100, 2604–2609 (2003)
13. Mempel, T., Henrickson, S.E., Von Andrian, U.H.: T-Cell Priming by Dendritic Cells in Lymph Nodes Occurs in Three Distinct Phases. Nature 427, 154–159 (2004)
14. Okada, T., Miller, M.J., Parker, I., Matthew, F., Krummel, M.F., Neighbors, M., Hartley, S., O'Garra, A., Cahalan, M.D., Cyster, J.G.: Antigen-Engaged B Cells Undergo Chemotaxis toward the T Zone and Form Motile Conjugates with Helper T Cells. PLoS Biology 3(6), 1047–1061 (2005)
15. Ma, B., Jablonska, J., Lindenmaier, W., Dittmar, K.: Immunohistochemical Study of the Reticular and Vascular Network of Mouse Lymph Node Using Vibratome Sections. Acta Histochem. 109, 15–28 (2007)
16. Seiden, P.E., Celada, F.: A Model for Simulating Cognate Recognition and Response in the Immune System. J. Theor. Biology 158(3), 329–357 (1992)
17. Efroni, S., Harel, D., Cohen, I.R.: Toward Rigorous Comprehension of Biological Complexity: Modeling, Execution, and Visualization of Thymic T-Cell Maturation. Gen. Res. 13, 2485–2497 (2003)
18. Beltman, J.B., Marée, A., Lynch, N.L., Miller, M.J., De Boer, R.J.: Lymph Node Topology Dictates T Cell Migration Behavior. J. Exp. Med. 204(4), 771–780 (2007)
19. Beltman, J.B., Marée, A., De Boer, R.J.: Spatial Modelling of Brief and Long Interactions between T Cells and Dendritic Cells. Imm. Cell Biol. (in press)
20. Stekel, D.J.: The Simulation of Density-Dependent Effects in the Recirculation of T Lymphocytes. Scand. J. Immunol. 47, 426–439 (1998)
21. Stekel, D.J., Parker, C.E., Nowak, M.A.: A model of Lymphocyte Recirculation. Immunol. Today 18, 216–221 (1997)
22. Beauchemin, C., Forrest, S., Koster, F.T.: Modeling Influenza Viral Dynamics in Tissue. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 23–36. Springer, Heidelberg (2006)
23. Murray, J.: Mathematical Biology. Springer, Heidelberg (1998)
24. Kallenberg, O.: Foundations of Modern Probability. Springer, Heidelberg (2002)
25. Blaschke, V., Micheel, B., Pabst, R., Westermann, J.: Lymphocyte Traffic Through Lymph Nodes and Peyer's Patches of the Rat: B- and T-Cell-Specific Migration Patterns Within the Tissue, and Their Dependence on Splenic Tissue. Cell Tissue Res. 282, 377–386 (1995)
26. Uematsu, T., Sano, M., Homma, K.: In Vitro High-Resolution Helical CT of Small Axillary Lymph Nodes in Patients with Breast Cancer: Correlation of CT and Histology. Am. J. Roentgenol. 176, 1069–1074 (2001)
27. Westermann, J., Puskas, Z., Pabst, R.: Blood Transit and Recirculation Kinetics of Lymphocyte Subsets in Normal Rats. Scand. J. Immunol. 28, 203–210 (1988)
28. Migration of So-Called Naive and Memory T Lymphocytes from Blood to Lymph in the Rat. J. Immunol. 152, 1744–1750 (1994)
29. Schwab, S., Pereira, J., Matloubian, M., Xu, Y., Huang, Y., Cyster, J.: Lymphocyte Sequestration Through S1P Lyase Inhibition and Disruption of S1P Gradients. Science 309, 1735–1739 (2005)

Revisiting the Central and Peripheral Immune System

Chris McEwan, Emma Hart, and Ben Paechter

Napier University, Edinburgh, Scotland
{c.mcewan,e.hart,b.paechter}@napier.ac.uk

Abstract. The idiotypic network has a long and chequered history in both theoretical immunology and Artificial Immune Systems. In terms of the latter, the drive for engineering applications has led to a diluted interpretation of the immunological models. Research inspired by theoretical immunology has produced compelling models of self-organised tolerance and immunity, but currently fail to have any practical engineering applicability. In this paper, we briefly discuss the engineering applicability of "self-affirming" idiotypic networks, leading to a suggestion that the "Third Generation" network models represent a way forward in this respect. Results obtained by implementing and extending a discrete model of this type of network suggest that the extended prototype is capable of two context-dependent modes of immune response, readily applicable to unsupervised machine-learning.

1 Introduction

The idiotypic network has received much attention in both theoretical immunology and Artificial Immune Systems (AIS) over many years. Leaving questions of full biological validity aside, it remains one of the most mature and cogent theories of how the immune-system may perform its seemingly cognitive tasks[1] without recourse to immunological or engineering anthropomorphism – arguably, an essential feature of any authentic "biologically-inspired" technique.

In an engineering context, any such technique must be able to accommodate some form of application-specific meaning, with subsequent structural and behavioural dynamics that are also meaningful in that context. This necessity for meaningfulness has, in the past, led to a diluted interpretation of the idiotypic network, built around traditional engineering measures and practices [12]. Recent research has shown that these interpretations, which account for a significant portion of existing AIS applications [16], fail to produce a network dynamic that could be considered "immune-like" [15] and further fail to produce solutions that could not already be engineered by traditional methods [1].

There is a certain tension between (and among) engineers and immunologists over the interpretation of the *Shape-Space* and measures of *Affinity* (e.g. see [9]). From a biological perspective, it is clear that such abstractions are crude at

[1] Such as discrimination, learning and regulation.

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 240–251, 2007.

best [8]. We would also question the utility of these abstractions in an engineering context: What can "immunological inspiration" contribute to the assumption that geometric proximity of points in an n-dimensional space reflects underlying relationships in the data? And how much of this inspiration is driven by immunological insight?

Research that continues to be inspired by the work of theoretical immunologists has produced some compelling models of self-organised tolerance and immunity. In [15], Hart and Bersini show that affinity measures defined in terms of Euclidean *similarity* produce networks that, because of their topological properties, fail to induce tolerant and intolerant regions of the space. Affinity measures defined in terms of Euclidean *complementarity* result in a bipartite topology that does indeed produce immune and tolerant regions of high and low field. However, it is not necessarily obvious how such *complementarity* can be applied in general; particularly in more abstract spaces, where the relationship between components is not strictly geometric.

Several other difficulties arise when trying to interpret and adapt theoretical models into an engineering context. In theoretical immunology, the issues surrounding these models has been discussed extensively (c.f. [5,2,3,4]). In the next section, we outline some of these problems from an engineering perspective.

1.1 Complements Will Not Get You Everywhere

As the complementary idiotypic network develops, it naturally produces regions of high field (leading to intolerance) and low field (leading to tolerance). B-Cells can only survive on the boundaries of such regions [15]. The B-Cell repertoire will therefore expand until it completely fills these boundary regions, resulting in an apparently stable network – the steady-state. In fact, the network has reached a ceiling it cannot progress beyond and the state is not steady at all: cell populations on these boundaries continue to proliferate unrestrained. From an engineering perspective, this presents no opportunity to extract meaning from the network's nodes.

Perhaps more awkward, is the overwhelming symmetry of the resultant compartments (for example, see [14]). It is difficult to justify an engineering task that involves stochastically partitioning the space into equal-sized regions of tolerance and immunity. Again, the opportunity to exploit the function of tolerance and immunity is limited, this time by lack of meaning in the network structure.

If a small amount of antigen are present during development, the network will attempt to organise around these antigen in a tolerant way. However, the success of this tolerance is entirely dependent on the location of the antigen: should one appear in the complementary region of another, then only one can be tolerated. Clearly, somatic antigen will be complementary, in order to form the physiology of the organism [21].

In the next section, we explore how Carneiro *et al.* approached such issues in the pursuit of biological meaningfulness.

2 Revisiting the Central and Peripheral Immune System

The Central and Peripheral Immune System was first suggested by Coutinho *et al.* [10,22]. Exploiting the strengths of both Burnet's Clonal Selection Theory and Jerne's Idiotypic Network Theory, they proposed a "Second Generation" immune network (SGIN) consisting of the Central Immune System (CIS) – a self-sustaining idiotypic network, providing a constant suppressive force on any potential response to antigenic patterns that have been integrated into its repertoire – and the Peripheral Immune System (PIS) – the majority of naive and less competitive lymphocytes responsible for classical adaptive immune response outside of the CIS influence. Only if PIS clones become integrated into the CIS repertoire can tolerance be achieved, and this tolerance is active and dominant (i.e. self-affirming) over the immune response [19].

B-Cell only models were never able to realistically induce these compartments and the idea remained largely descriptive until Carneiro *et al.* were able to break the symmetry of these models by including the co-operative interactions with T-Clones [20]. These two modes of B-T interaction are presented in Fig. 1.

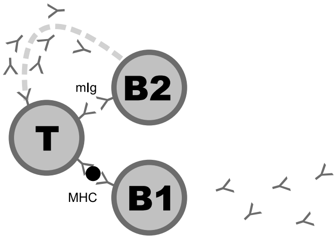


Fig. 1. Two modes of B-T interaction. (i) B1 perform MHC-Presentation. The resulting Ig having little affinity with the TCR and both clones proliferate in a positive feedback loop (ii) B2 interacts via direct BCR recognition of the TCR. The resulting secreted Ig can directly suppress T in a stable, negative feedback loop.

Herein lies the crux of the "Third Generation" Immune Network (TGIN) model: B-Clone activation (and thus secretion) is constrained by available T-Help. Available T-Help is constrained by the antigenic stimulation and the suppressive effects of secreted immunoglobulin. The resultant waves of idiotope-anti-idiotope etc. form a self-regulating negative feedback loop, during which only the most competitive B-Clones can fully exploit T-Help – relegating weaker clones back to the PIS and "focussing" the CIS repertoire to best reflect its antigenic stimulus [6].

The original TGIN model implements a probabilistic, *multi-point* shape-space to reflect the "continuous epitope" nature of ligand binding. In [8] the authors make a convincing case for the inadequacy of single-point shape-spaces and the inherent difficulty of extracting generalised affinity measures from empirical analysis of protein-binding data. We find these comments ring true in a machine-learning context, and consider the capacity of the model to perform without any implied underlying topology an opportunity for engineering applications of the idiotypic network that do not simply re-interpret classical data-analysis and clustering methods.

However, the motivation for the TGIN was the mechanism of distinct CIS/PIS compartments during the *ontogenesis* of the immune-system; when the only antigenic presence is somatic antigen. In general, it is the *ontogeny* of the system in the presence of pathogenic and benign antigen that is of interest – particularly in the case of on-line, unsupervised learning. Over the system's ontogeny, clone lifecycle and interactions become much more exquisite. The remainder of this paper addresses original modelling simplifications that fail to hold in this situation; necessary modifications; and any additional considerations that must be accounted for.

3 Implementation of a Third Generation Network Model

In principle, the TGIN model is a straight-forward extension of Varela's earlier model [22]: it adds T-Cells and differentiates between induced and activated B-Cells. In fact, the model is complex and suffers from many parameters and simplifying assumptions. We only sketch an outline here to avoid excessive commentary and direct the reader to [6] for the full details.

Both B-Clone and T-Clone population dynamics are described with differential equations of the form:

$$\frac{dx_i}{dt} = (-X_{decay} \times x_i) + (X_{prolif} \times x_{i.activation}) \tag{1}$$

i.e. clones decay exponentially by a constant decay rate and increase proportionally to their level of activation. For T-Clones, activation is a function of the stimulatory effects of Antigen and the suppressive effects of anti-TCR Immunoglubulin. For B-Clones, activation is a function of (i) the number of induced cells (itself a function of recognised Immunoglobulin and Antigen) and, (ii) the competitive fitness of the clone to exploit available T-Help (a function of the B-Clone's interaction strength with the T-Clone).

Transforming the differential equations to difference equations was done using Euler's Method:

$$\frac{dx}{dt} = f(x, \cdots) \approx x_{t+1} = x_t + \Delta f(x_t, \cdots) \tag{2}$$

where Δ approaches zero. In all cases it was set to 0.1. Experimentation with this parameter had little qualitative effect on the dynamics, other than adjusting the window of time for new clones of low population to survive unstimulated. In all cases, parameters remain as in the original simulations.

In its original form, the model is computationally prohibitive. We make extensive use of caching intermediate non-linear terms between time-steps. Technically, this introduces some slight synchronicity into the model, although no qualitative difference in the dynamics was observed. The algorithm now runs comparatively with other SGIN models.

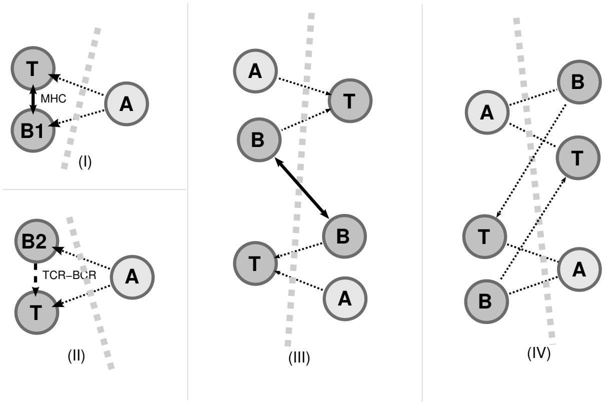


Fig. 2. The TGIN prototypical models: (I) and (II) represent the two modes of B-T Co-operation discussed earlier. (III) shows the mutual induction of B-Clones via their (antigenic) Ig and the suppression of independent activated T-Clones. (IV) shows the mutual suppression of each others co-operating T-Clone. The bold dashed line represents a hypothetical boundary between complementary regions.

3.1 An Augmented Prototype

In [6], Carneiro *et al.* present four *prototypical* models, intended to illustrate minimal but representative interactions in the full simulation (see Fig. 2). In studying these we note that, with the exception of one, all are directly interpretable in a complementary shape-space. The exception is prototype II, which fails to translate as it requires a triangular affinity relationship, which is generally impossible in a complementary space. However, given the other prototypical modes of interaction, it is possible to augment this second prototype to achieve a similar mechanism (see Fig. 3).

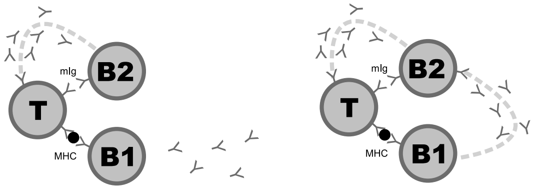


Fig. 3. Translating Carneiros model to a complementary shape-space. The original model (left) is slightly modified to explicitly incorporate the to-and-fro interactions between idiotype and anti-idiotype. This defines the minimal stable network and is our augmented prototype. (right) Ig secretions from the antigen-specific cell invoke a second wave of immune response. The *anti-B-Clone*, induced by the Ig, necessarily has affinity with the TCR. Once activated via T-Clone co-operation, anti-TCR Ig secretion will suppress the T-Clone and indirectly suppress the original antigen-specific clone, thus stabilising the network.

3.2 Antigen Lifecycle

The most obvious departure from the original model is the time-varying presence of antigen, their opsonisation and its dynamical effect on the network. To remain as faithful to the original model as possible we apply the same function as T-Clone suppression to antigen population dynamics, as both are suppressed by soluble Ig.

$$\frac{dA_k}{dt} = -A_{opsonise} \sum_{i=0} affinity(A_k, F_i) \times F_i \tag{3}$$

In the experiments discussed here, antigen concentrations are either decreasing or constant. Antigen do not decay naturally. $A_{opsonise}$ represents the success rate of soluble Ig (F) in opsonisation. We explored the ranges of 0.1, 0.5 and 1.0 for this parameter.

3.3 T-Cell Lifecycle

In the original model, the only suppressive force on T-Clones is from anti-TCR immunoglobulin. In fact, the behaviour is more subtle: given the original simplifying assumptions, a solitary T-Clone will proliferate exponentially without any other influence acting upon it. Without tracing the full dependencies that result in this behaviour, we state that this is a result of the simplification that a T-Clones level of stimulation can be replaced with a unitary constant [7]. In the original model, this simplification could be justified. In our model, it fails to hold and the T-Clones invariably explode.

Leon *et al* [18] modelled T-Clone stimulation as a log-normal curve of the antigen concentration. While we do not run under the same parameters as that later model, we have verified that using such a function produces qualitatively better T-Clone dynamics over the ontogeny of the system.

However, there is now another scenario to contend with: T-Clones can die, either by excessive anti-TCR Ig suppression or lack of antigenic stimulation. In the original model, for each antigenic input a dedicated T-Clone is created with maximal affinity. If this T-Clone dies in the current model, then the immune-system becomes blind to the antigen – no induced B-Clone can receive T-Help to further secrete immunoglobulin. While this is certainly a crude form of tolerance, we consider the mechanism undesirable as it results in a system that cannot adapt over time. To offset this effect, we consider the random creation of T-Cells in the meta-dynamics of the model.

3.4 B-T Interaction Strength

In a maximal simplification, the original model assumed that the interaction strength between B-Clones and T-Clones could be replaced with a unitary constant, regardless of their mode of interaction [6]. Given that B-T interaction is the key to repertoire selection – and thus, exploitable meaning – this simplification is highly undesirable. A side-effect of employing a constant interaction strength is that all B-Clones receive some positive interaction with T-Clones; achieve a level of activation; secrete; and are artificially sustained in the idiotypic repertoire. This is inevitable as a constant term does not take account of the possibility of no interaction. Carneiro *et al.* do however provide equations for the two modes of interaction:

$$S_{BCR}(B_i, T_i) = affinity(B_i, T_i) \tag{4}$$

$$S_{MHC}(B_i, T_i) = s \cdot \sum_{k=1} affinity(T_i, A_k) \cdot affinity(B_i, A_k) \cdot A_k \tag{5}$$

Although the latter equation was justified on biological grounds, we found the antigen concentration term produced unintuitive behaviour. In the case of constant (somatic) antigen, MHC-Presentation is exponentially more competitive – exactly the opposite behaviour of that desired for a suppressive tolerant response. In the case of variable antigen, the MHC interaction-strength decreases to zero as the antigen population drops, leaving only the BCR-TCR interaction in a possible stable-state – again, the opposite type of‘ “memory” than would be expected for immunity. The constant scaling-factor s is less meaningful under variable antigen concentrations[2].

Additionally, how to resolve which equation to apply for a specific clone was not elaborated and such branching is inherently problematic for static numerical analysis. The complementary space provides a clear manner to make this distinction:

$$S = max(affinity(T_i, B_i), affinity(T_i, B_{i.complement})) \tag{6}$$

If a T-Clone has high affinity to the complement of a B-Clone, both recognise overlapping regions of the space and their respective receptors have a low affinity by virtue of their similarity – interaction is via MHC-Presentation. In the case that both appear in each other's recognition region, their receptors have high affinity, they cannot recognise similar antigen and interaction is via BCR-TCR binding. The remainder of the space has no possible interactions. While a crude simplification of the biology, Eq. 6 does produce desirable values between 0.0 and 1.0, keeping MHC and BCR interactions in the same scale and also in scale with the original unitary simplification. As a result, the constant scaling-factor s can be meaningfully re-employed to weight MHC interactions.

4 Simulation Results

Here we present simulation results of the qualitative behaviour of our extended prototype. The combinations of configurations explored is quite large. Given the forum, we only present major results and outstanding issues; saving a detailed analysis for a more appropriate text.

[2] In order to induce tolerance, interactions via MHC must be less effective than TCR-BCR interactions. This is biologically plausible as there are significantly more BCR than MHC-Peptide complexes on the cell surface. s represents this proportionality.

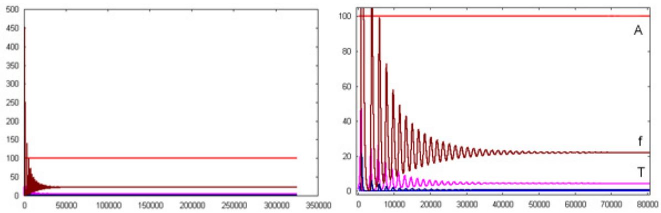


Fig. 4. Stabilising network behaviour in the augmented prototype: Given a constant antigenic presence, the clone populations reach a fixed point between T-Clone activation and anti-Ig suppression. The anti-Ig (f) offsets the T-Clone activation and actively suppresses responding B-Clones.

The Augmented Prototype. Here we validate that the minimal network does indeed stabilise to an appropriate configuration (see Fig 4). We note that this stabilising process is quite prolonged, however it was our intention to avoid adjusting the original model parameters until we had validated our changes.

Repertoire Management. One of the key benefits of the TGIN is that during an immune response, the CIS repertoire is pruned to only the most competitive cells. This avoids the tendency to "repertoire completeness" and is the essential distinction between the CIS and the PIS. Following the original analysis [7,19], we plot the expansion of the idiotypic repertoire over time (see Fig. 5).

For comparison, we plot Bersini and Hart's algorithm in (a). Without antigen, the repertoire expands and settles at a massively inflated value. This is the ceiling of repertoire completeness discussed earlier – there is no more room in the space for B-Clones. With antigen, the repertoire overflows the boundaries and completely fills the tolerant regions, ultimately populating half of the space – $Pr \times Nb \approx 160$ (not shown). In (b) we plot the extended TGIN model, showing the characteristic repertoire expansion then compression to ≈ 1 per antigen for all values of $s < 1.0$. In (c) we plot the same extended model running

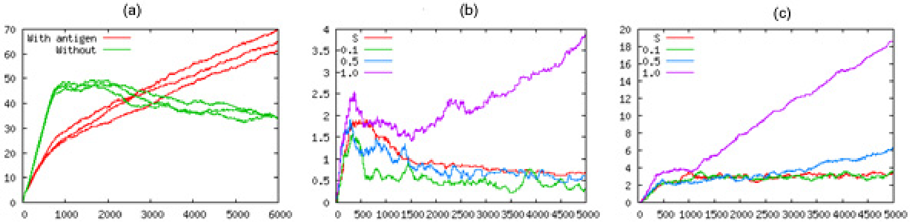


Fig. 5. Expansion of the idiotypic repertoire over time. The x-axis represents time. The y-axis represents $NB \times Pr$ – the number of B-Clones times the probability of recognising another clone. The number of presented antigen (Na) is set to $Na \times Pr \approx 1.0$. (a) plots several runs of Hart & Bersini's algorithm with and without antigen. For (b) and (c), we plot our TGIN implementation for different values of the scaling factor s.

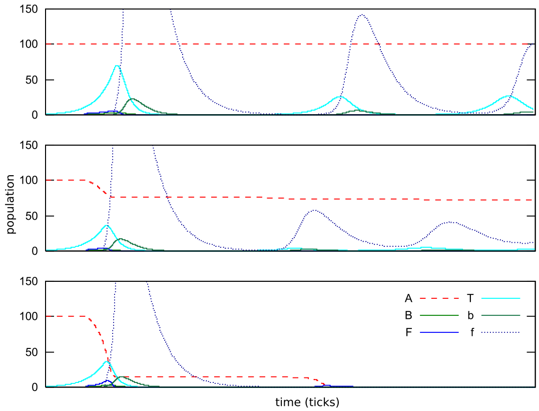


Fig. 6. Effect of antigen opsonisation on the extended model: (Top) Constant antigen and the typical stabilising oscillations between the activated T-Cell and the suppressive anti-Ig. (Center) At an opsonisation rate of 0.1 we see an initially strong auto-immune response followed by progressively weaker intermittent responses. At an opsonisation rate of 1.0 we see an intolerant response – tolerance is attempted but ultimately fails.

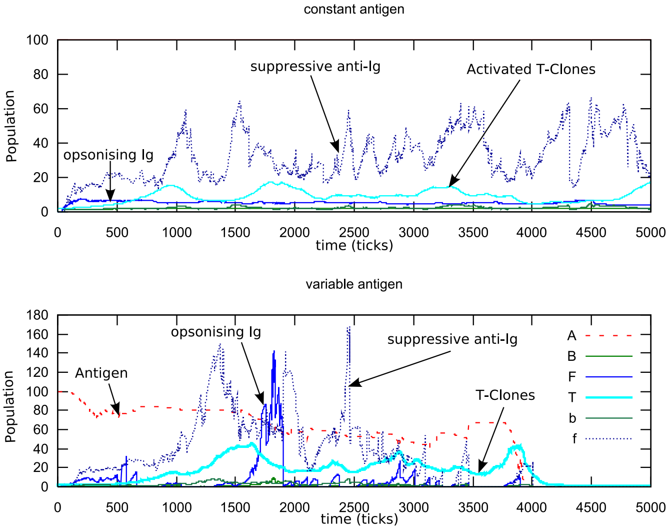


Fig. 7. Two modes of antigenic response during the full ontogeny. For somatic antigen, responding clones are kept under a blanket of suppressive anti-TCR Ig. For non-regenerating antigen, stable tolerance never gains full dominance and can only regulate the inevitable immune response. Crucially, there is no mechanism for memorising cleared antigenic disturbances, resulting in the collapse of the antigen-specific clones.

with additional T-Clone meta-dynamics which (because more than one T-Clone can respond to an antigen and thus sustain B-Clones) accommodates a slightly elevated, but still low and stable level of B-Clones for all $s < 0.5$.

Two Modes of Response. Figure 6 plots three different responses under different opsonisation rates. Of particular interest is the center plot, showing an antigen population approaching a fixed-point with the stabilising augmented prototype. This suggest that a correctly parameterised network may not only be capable of tolerance but also maintaining somatic antigen populations, as has often been suggested by Cohen, Coutinho and others.

When translating to the ontogeny simulation (Fig. 7), qualitative analysis and visualisation becomes significantly less intuitive. Here we present aggregate population plots of the extended model running with full meta-dynamics. Even from such a coarse-grained view, it seems clear that the behaviour is qualitatively similar to the augmented prototype in the case of constant antigenic presence. In the case of variable antigen, the behaviour is much more erratic as the clones compete to fulfill their function.

5 Discussion

We have presented a prototype immune-network model that appears capable of two context-dependent modes of antigenic response. However the context of response is not cognised by the network *per se*, but a result of a regenerating (effectively constant) "self" that is able to reach a stable equilibrium despite brief autoimmune episodes. The idiotypic network provides the regulatory mechanism to minimise the effects of any single autoimmune response and maintain a suppressive field over autoreactive clones.

Under the complementary shape-space the "Central Immune System" is not a fully connected idiotypic network, but rather a federation of isolated networks. This is a necessary outcome of optimising the repertoire in a single-point space, and somewhat contradicts the original idea of tolerance as an emergent global property of the network [10]. As a result, a key mechanism missing from this single-point model is *memory.* For an idiotypic network to be maintained, secretion must continue, which fails if either the Antigen or T-Clone are removed from the network. In the original TGIN model, both were impossible.

These issues may only affect single-point shape-spaces, where connectivity depends on the overlapping recognition regions of redundant clones. In a multi-point space, clones can receive stimulation from several locations and do not need to be contiguous to maintain connectivity and co-stimulation. However, memory is not the immediate concern of a self-affirming network, and furthermore, such idiotypic memory has little biological support. There are other elegant mechanisms that can be exploited (c.f. [13]) without the computational burden of maintaining a growing network of co-stimulatory dominant and recessive clones.

This paper re-introduced the TGIN in a single-point space, to place it in context with existing AIS research. Future applications of the model will remove this topological limitation, in which case we will be better able to appreciate the full relation between topology and function. Even under such constraints, the current model is free from anthropomorphism and directly interpretable in a machine-learning context: *habituation.* If there is any credibility in the idea of the immune system as a cognitive network, we consider this an appropriate starting point for integrating meaningful *association* and *memory* into networks in a multi-point space. We consider this future work.

A self-affirming network is an entirely different beast from early idiotypic models in AIS proposed by e.g. [17,11]. The latter are loosely concerned with immunological memory and assume that the topology of the idiotypic network represents some underlying topology in the antigenic data. The TGIN is concerned with the functions of immunity and tolerance. It is not bound to any specific topological space and does not *cognise-by-clustering*. As such, we feel it provides a compelling opportunity for tackling problem domains beyond vectorial data-analysis.

References

1. Bersini, H.: Self-assertion versus self-recognition: A tribute to Francisco Varela. In: Timmis, J., Bentley, P.J. (eds.) Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS), pp. 107–112. University of Kent at Canterbury, University of Kent at Canterbury Printing Unit (September 2002)
2. De Boer, R.J., Hogeweg, P.: Unreasonable implications of reasonable idiotypic network assumptions. Bulletin of Mathematical Biology 51, 381–408 (1989)
3. De Boer, R.J., Perelson, A., Keverekidid, I.G.: Immune network behaviour - i. from stationary states to limit cycle oscillations. Bulletin of Mathematical Biology 55, 745–780 (1993)
4. De Boer, R.J., Perelson, A., Keverekidid, I.G.: Immune network behaviour - ii. from oscillations to chaos and stationary states. Bulletin of Mathematical Biology 55, 781–816 (1993)
5. Carneiro, J.: Towards a Comprehensive View of the Immune System. PhD thesis, University of Porto (1997)
6. 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)
7. 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)
8. 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)
9. De Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, London (2002)
10. Coutinho, A.: Beyond clonal selection and network. Immunological Reviews 110, 63–88 (1989)
11. de Castro, L.N., Zuben, F.J.V.: Data-Mining: A Heuristic Approach, chapter aiNet: An Artificial Immune Network for Data Analysis, pp. 231–259. Idea Group Publishing, USA (2001)
12. Galeano, J.C., Veloza-Suan, A., Gonzalez, F.A.: A comparative analysis of artificial immune network models. In: GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation, pp. 361–368. ACM Press, New York, NY, USA (2005)

13. Garret, S., Robbins, M., Walkerand, J., Wilson, W., Aickelin, U.: In Silico Immunology, chapter Modelling Immunological Memory. Springer, Heidelberg (2006)
14. 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)
15. Hart, E., Bersini, H., Santos, F.: Tolerance vs intolerance: How affinity defines topology in an idiotypic network. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 109–121. Springer, Heidelberg (2006)
16. 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, pp. 29–42. Springer, Heidelberg (2005)
17. Hunt, J.E., Cooke, D.E.: Learning using an artificial immune system. Journal of Network and Computer Applications 19, 189–212 (1996)
18. Leon, K., Carneiro, J., Perez, R., Montero, E., Lage, A.: Natural and induced tolerance in an immune network model. Journal of Theoretical Biology 193, 519–534 (1998)
19. Stewart, J., Carneiro, J.: Artificial Immune Systems and their Applications, chapter The central and the peripheral immune system: What is the relationship? pp. 47–64. Springer, Heidelberg (1998)
20. Stewart, J., Coutinho, A.: The affirmation of self: A new perspective on the immune system. Artificial Life 10, 261–276 (2004)
21. Takumi, K., De Boer, R.J.: Self assertion modeled as a network of multideterminant antibodies. Journal of Theoretical Biology 183, 55–66 (1996)
22. Varela, F.J., Coutinho, A.: Second generation immune networks. Immunology Today 12(5), 159–166 (1991)

Topological Constraints in the Evolution of Idiotypic Networks

Emma Hart[1], Francisco Santos[2], and Hugues Bersini[2]

[1] School of Computing, Napier University
e.hart@napier.ac.uk
[2] IRIDIA, Universite de Bruxelles
{bersini,fsantos}@ulb.ac.be

Abstract. Previous studies have shown that there is an intricate relationship between the topology of an idiotypic network and its resulting properties. However, empirical studies can only be performed by preselecting both a shape-space and affinity function. This introduces a number of simplifications into any model and makes it subsequently difficult to abstract the underlying contribution made by the topology from the particular instantiation of the model. In this paper, we introduce the concept of the *potential network* as a method in which abstract network topologies can be directly studied which allows us to bypass any definition of shape-space and affinity function. By using ideas from complex network theory to study a variety of homogeneous and heterogeneous potential networks, we show that bi-partide and heterogeneous topologies are able to tolerate antigens in certain regions, where as those showing high levels of clustering are unable to do so. It is also shown that the *equilibrium* topology resulting from traditional immune dynamics depends dramatically on the potential topology of a network.

1 Introduction

In recent years, it has been experimentally observed that real-world biological, social and technological networks are not structured in a random way [1]. Instead, most of these networks are organized in such a way that a few nodes are able to interact with a many others, whereas many others only interact with a few. The extreme case is often referred to as a scale-free network in which the degree distribution follows a power-law [3]. However, other configurations showing lower levels of heterogeneity are also common [2]. Dissections of real world networks have produced evidence for single-scale networks, characterized by a fast Gaussian decaying tail in the degree distribution, broad-scale distributions, defined by a power-law with an abrupt truncation for large connectivities [21] and, finally, the previously referred scale-free class. The ubiquity of such classes of networks raises many questions of which one is on the origin of these topological properties. Moreover, in the context of understanding complex social and biological phenomena, it is useful, and in many cases mandatory, to understand the topology of the underlying networks of interactions [1,11,4]. The

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 252–263, 2007.

global properties of a system rely extensively upon the underlying network and different dynamical outcomes emerge from different topologies [16,17,12,19].

One of the major goals of modeling in immunology, which we shall focus here in a very minimal fashion, is to deduce macroscopic properties of the system from the dynamics of its elementary components. Thus, nodes in a network can be considered to correspond to the different cells that constitute the basis of the immune machinery, and a link in a network is analagous with the reaction obtained when one antibody (or antigen) reacts with complementary/similar receptors of another cell. When such high-level abstractions of the immune system are considered, analogies with the nervous system become apparent. While in the nervous system we can identify a sensorial part and a motor part, in immunology we can also assign two main functions to the entire system: *recognition* and *effector* function. However, due to the great diversity of possible cells and to the fact that only one in 10^5 cells reacts to a specific cell, it is hard to extract experimentally the real topology associated with an immune network, and it therefore remains a completely open question as to how to determine a realistic topology for such networks.

One of the most common approaches in theoretical studies has been to retrospectively examine the topologies that arise in networks after specifying a way in which to define cells and a specific manner in which cells can interact to form a network. This was greatly facilitated by the introduction of the concept of *shape-space* by Perelson in [18] which offered the possibility of representing cells in some low-dimensional space in which properties such as mututal affinity and similarity of cells could be derived from the relative positions of points in the space. Mathematically this can be expressed as $c_{ij} = f(x_i, x_j)$ where c_{ij} is the affinity between two molecules i and j, X_i and X_j are N-dimensional vectors representing the position of the molecules in shape-space and f is an appropriate function [8]. The function f therefore defines an *affinity-matrix* which specifies all interactions among cells. Several models ensue; it is often assumed that c_{ij} can be calculated by a simple Gaussian metric based on the distance $||A-B'||$ where B' is the symmetrical image of a point B through some point in shape-space [10,6]; Hamming distance is substituted in a bitstring shape-space; Carneiro *et al* define a random affinity-matrix in [9]. However, all of these methods suffer from a major drawback. In order to be realistic, the dimensionality of N needs to be very high and it is likely that f is irregular and discontinuous [8]. Indeed Carneiro *at al* even go as far as to suggest that *...the danger is that heuristically stimulating visions of the organisation of the immune system based on the shape-space concept may be illusions based on little else than wishful thinking.*

Therefore, in this paper we present an alternative approach to studying idiotypic immune networks; instead of trying to understand the topology which results from specifying a shape-space and affinity function, we directly study the topologies of *abstract potential networks* on which *effective networks* can evolve, and apply the complex networks' formalism to the study of such networks. Ultimately, we show that the choice of affinity function which results in the potential network formation is far from being neutral. The remainder of the paper is organised as follows. We first discuss the topologies of some common shape-spaces and affinity functions, summarising the results of the previous empirical studies that we have publihsed. This is followed by an introduction to the concept of the *potential network*. We then present results obtained by analysing the dynamics of an immune model on homogeneous and heterogeneous potential networks, and show that the topology of a network offers an explanation for one of the observed functions of an immune network, namely that it can separate zones of tolerance from zones of immunisation.

2 The Topologies of Some Common Shape-Spaces

2D shape-space One of the most commonly adopted shape-spaces is a 2D shape-space model, firstly proposed by one of the authors in [5] and subsequently adopted in further work [13,14] which lends itself naturally to affinity measures based on Euclidean distances. The network of all possible interactions is defined on a 2D integer-grid of dimension X, Y. A cell is specified by a position (x, y) on the grid. The potential network therefore consists of a possible $X \times Y$ cells. Cells can be considered as connected nodes on a graph if one cell is *stimulated* by another cell. The manner in which one cell stimulates another depends on the affinity function defined. If affinity is defined as complementary, then a cell A stimulates another B if B lies within a circular region of radius r centered on the point $(X-x, Y-y)$. On the other hand, if affinity is defined between *similar* cells, then A stimulates B if B lies within a circular region of radius r centered on A itself. In this case, each node is a point in a 2D plan, which can *potentially* interact with all other cells characterized by a position which is complementary or similar to its own position. This potential network is almost completely homogeneous in that every cell in the network can potentially stimulate exactly the same number of other cells. (The exceptions to this are cells lying closer than r to the edges of the 2D space whose recognition region will lie partially outside the 2D grid. No wrap-around effects were implemented). The *average degree* of any cell in the network is defined by the area of the stimulation zone surrounding the cell, i.e. r. The *potential degree* of any cell is therefore the maximum number of other cells to which it can potentially connect, governed of course by the area defined by r and is equal for all cells.

However, depending on the affinity function used, in previous studies we showed that different potential topologies naturally emerge [15] and suggested that the network behaviour could be explained by the existence of certain motifs in the topology; in a network defined by a similarity based affinity function, then a number of *triangles* can exist, as if A is close to B, which is also close to C, then C can also be close to A. On the contrary, if a complementary definition of affinity is used, then in the majority of the space, is impossible to obtain such triangles: if A stimulates B and B stimulates C, then A cannot stimulate C because A is similar to C. (The exception is cells lying in the centre of the space where it *is* possible to obtain such a motif). Formally, this feature appears naturally if we analyse the network topology using the standard tools of complex network theory. If one compares the cluster coefficient in the first case, it is obvious that it will be relatively high, while in the second case it must be (almost) zero.

Bitstring Shape-space. In [15], we also examined the emergence of networks in a bitstring shape-space. Here, instead of a point in a plane, each cell is now identified by a binary bit-string of N bits and the a cell i will stimulate another cell j if the Hamming distance among them is higher than a certain threshold T. This value T plays an equivalent role of the parameter r in the 2D shape-space model in determining the number of possible interactions any cell may take part in. Regardless of the value of T, a homogeneous potential network will always be obtained, where every cell can potentially interact with the same number of cells, which may be present in the system or not. However, the parameter T also plays another important role: turning once again to complex network theory, we see that T influences an important topological feature, the global cluster coefficient of the network. This is shown empirically in figure 1 in which the average cluster coefficient of the potential network in which all potential nodes exist is calculated for bitstrings of length (9,10,11). Furthermore, it is trivial to prove that in fact the cluster coefficient of any potential network formed from bistrings of length N will be non-zero in any case in which $T < 2N/3$. The values plotted represent the maximum value of the cluster-coefficient of these networks; in reality, not all nodes will exist in the effective network and therefore empirically measured cluster coefficients are likely to be less than these values.

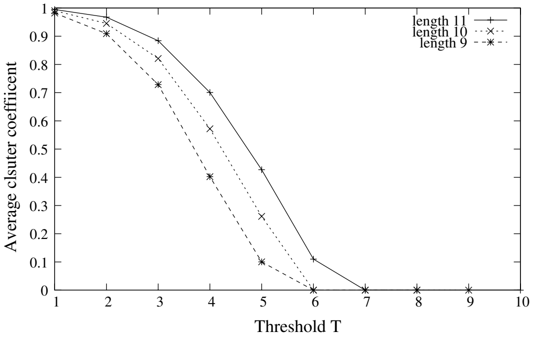


Fig. 1. The graph shows the average cluster coefficient in a potential network created by a bistring of length L with increasing threshold T

In [15] we showed empirically that the cluster coefficients of networks evolved in Hamming and 2D shape-spaces do have radically different values, and tentatively suggested that this property was responsible for the ability of the network to tolerate or reject certain classes of antigens. However, these models clearly represent a gross simplification of real interactions between immunological cells and makes it difficult to come to compelling conclusions. However, focusing explicitly on the topology of the networks formed as a result of interactions between cells due function suggests an alternative experimental approach, which is not dogged by the problems described in section 1. This is elaborated in the next section in which we show how standard immune dynamics can be studied on abstract homogeneous potential networks with zero and non-zero values of cluster coefficient respectively, providing further evidence for the impact of this topological feature on network functionality.

3 Immune Dynamics and Potential Networks

Let us assume that there exists a plausible representation $\mathcal{R}$ in which cells interact via a realistic affinity function $\mathcal{A}$. This gives rise to a network N with a particular topology which implicitly defines a shape-space (by defining a certain topology), and affinity function, and even the stimulation area of any cell (by defining the individual potential degree). Most importantly, this allows us to bypass any definitions of these parameters and directly study the effect of network topology on the emergence of tolerant zones within a shape-space.

Assume a potential, homogeneous graph G which defines a potential network. We consider graphs in which a cell X stimulates another cell Y if X has concentration different from 0 and if X has a link with Y in the underlying potential network. All connections are assumed to have the same weighting. The stimulation $s_{(X,Y)}$ received by a cell X from Y is zero if X does not have a link with Y in the potential network, and is equal to $\alpha * C_Y$ otherwise, where α is simply a pre-defined constant and C_Y is Y's concentration. Depending on the total amount of stimulation received by X, its concentration will either increase or decrease, just like in the models presented in [15] and earlier by [5,20]. If the total stimulation is between a lower and a higher stimulation thresholds its concentration will increase by one, and decrease also by one unit otherwise. Every 1000 iterations of the algorithm, a set of 50 antigens is added to the simulation. Each time, the set is generated completely at random, and excludes duplicates[1]. For a potential network of N nodes, at each time-step the algorithm runs as follows:

- Introduce a new antibody cell by randomly choosing an empty node ($C_i = 0$) from the potential network and assign it a concentration $C_i > 0$.
- Calculate the total stimulation of each cell with non-zero concentration, summing the stimulation received from each first neighbour in the potential network.
- Update the concentration of each cell as described above.
- Every 1000 iterations, the number of antigens that remain in the population of non-zero concentration cells is recorded. These cells are then removed and a new set of antigens then introduced.

[1] Extensive experimentation showed that the results did not depend in any way on the manner in which antigens were added.

4 Homogeneous Potential Networks

We first study the topological effects on the emergence of tolerance, assuming a homogeneous graph in which all nodes share the same potential degree. To shed further light on the proposed contribution of the triangular motif to the functionality of networks in terms of their ability to produce distinctive and separated tolerant and intolerant regions, we study two kinds of regular graphs: a regular ring of size N and a bi-partide regular network made of two rings of size $\frac{N}{2}$ (see figure 2) with an average degree $< k >$. The former case corresponds to an affinity function based on the similarity between cells. This creates a potential network where triangles or loops are often present. In the latter case, we create a regular network that can pictured as two parallel rings of nodes — ring A and B. Nodes belonging to the first one ring can only be connected with nodes of ring B and vice-versa. For instance, for an average degree of 4, a node i belonging to the ring A will be connected to the node at the position $i-1$, $i-2$, $i+1$ and $i+2$ of the ring B. In this way, we produce the bi-partide equivalent of a regular network, with exactly the same spatial constraints and average degree, but implicitly defining two groups of interacting cells in a bi-partide fashion. The cluster coefficient will be trivially equal to zero in this case, where in the normal regular network will be given by $\frac{3z-6}{4z-4}$, with z standing for the average degree of the network [11]. This bi-partide regular graph corresponds to the complementarity affinity function, which has been shown to give rise to two perfectly distinct areas in terms of tolerating antigens, contrary to the similarity rule [15].

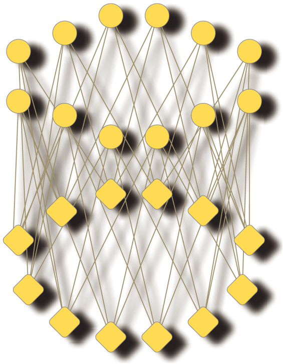


Fig. 2. This scheme illustrates a bi-partide equivalent of a regular graph corresponding to the complementarity affinity rule. In this example the average degree is 4. The ring made of circles on the top is the complement of the ring formed of square on the bottom.

4.1 Results

The results for this simplified model using homogeneous networks are shown in figure 3. In both cases, we consider potential networks of 10^4 nodes in order to provide comparable results to those obtained empirically in [15]. These results correspond to an average over 50 runs, obtained using an initial antibody concentration and initial antigen concentration of 200. In the upper panel we show the percentage of tolerated random antigens as a function of the average degree of the underlying regular network for two types of regular networks: a regular ring and bipartide regular ring. The middle panel shows the dependence of the equilibrium size of the network in the potential average degree, and the lower panel represents the relationship between the potential average degree and the effective average degree of the nodes with concentration different from 0.

These results can be compared directly to previous results obtained in 2D ([15]) in which the recognition radius of a cell was plotted against % tolerated antigens — the potential average degree is topologically equivalent to the radius of stimulation. They show that the existence of a bi-partide topology defining the set of all possible/potential interactions promotes the emergence of high levels of tolerance for most of the values of $< k >$. This result corroborates with results obtained with a 2D-shape space model, validating the simplification and abstraction introduced here.

Figure 3 shows three regions (in terms of $< k >$) limited by abrupt transitions. The first one occurs when the potential network doesn't provide enough stimulation to maintain the majority of the antibodies inside the window range (i.e. there is low potential average degree). For a moderate level of potential links, a second type of behaviour emerges. Here the potential network offers the ideal conditions for the emergence of a self-sustained system of antibodies, avoiding both under and over stimulation. In this range the effective size of the network increases together with the effective average degree. Finally, when the potential average degree becomes too high, the majority of nodes are able to become over-stimulated. Under these conditions, the equilibrium size of the network collapses together with the effective average degree. This leads to the first conclusion of this paper: the existence of a non-zero value for the cluster coefficient of a network acts as an important deterrent for the emergence of distinct tolerant zones in the shape space, at least when homogeneous potential networks are considered.

5 Heterogeneous Potential Networks

The previous paragraphs have used two instantiations of a homogeneous network model coupled with various affinity measures, and in abstract *potential* networks. However, this work, as in the majority of previously published models of idiotypic models, has made the assumption that the affinity function is defined in such a way that every cell type has intrinsically the same number of potential stimulation partners. This disregards a plethora of recent results in the area of complex biochemical networks showing that the majority of the real-world network do not share this homogeneous feature. In fact, it has been extensively shown that most of the biological networks are intrinsically heterogeneous, where

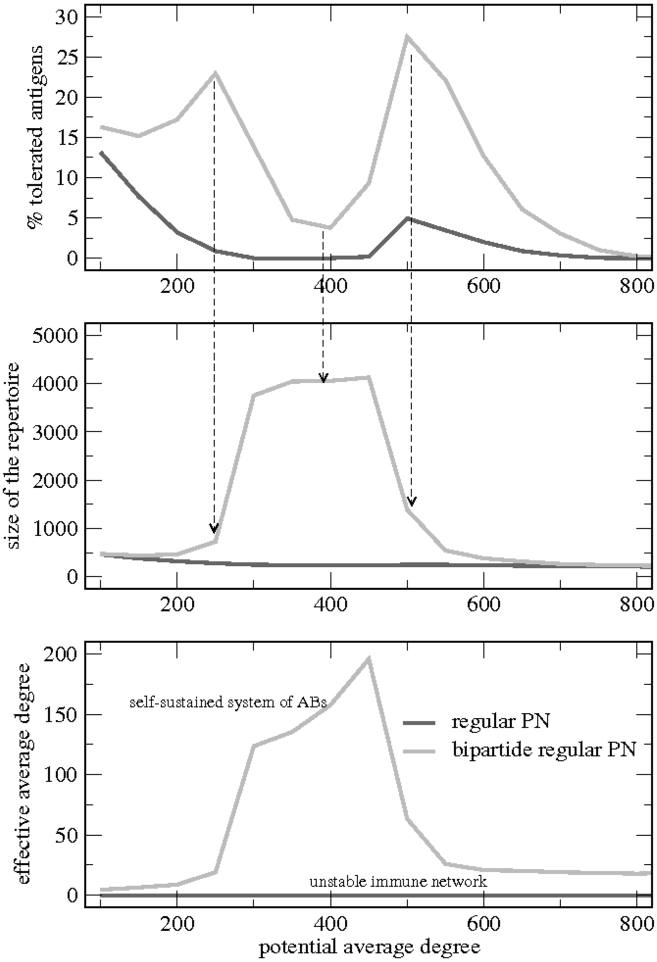


Fig. 3. Results obtained using homogeneous potential networks. The graphs show results obtained in regular and bipartide potential networks as the average degree of the potential network is varied. The graphs show the corresponding change in % of tolerated antigens, network size and effective average degree.

some nodes, considering their intrinsic chemical properties, are able to stimulate a large number of cells, contrary to others that stimulates only a few number of cell types. The first ones are naturally *made* to connect to a large number of other cells - they are *natural hubs* [7]. Moreover, it has been shown that most of the biochemical networks can be characterised with broad-scale degree distribution or even by a scale-free degree distribution, recently popularised by their remarkable robustness properties. Therefore, although our results provide a novel connection between the topological characteristics of a network (resulting from the affinity measure) and it's ability to tolerate or immunise antigens, a natural extension to this work would therefore be to consider the emergence of effective networks on heterogeneous potential graphs, and furthermore, potential graphs exhibiting scale-free (or nearly scale-free) degree distributions.

Heterogeneity in a network may offer the potential for certain types of antigen (depending on their degree and position in the global interaction network) to become topologically protected against antibodies, which are not able to destroy them, therefore increasing the global level of tolerance. Moreover, heterogeneity effects also play an important role in determining the equilibrium size of the repertoire, which in turn influences the capacity of an idiotypic immune network to tolerate antigens.

5.1 Results

We use the same immune model as before, but consider two heterogeneous potential graphs; a scale-free potential network (similar to the WWW model proposed by Barabasi [3], and a bipartide scale-free potential network, again both containing 10^4 nodes. Potential graphs of varying average degree (0 to 1500) are generated at random, and then the effective network evolved in each graph. Each experiment is repeated 20 times and the % of tolerated antigens averaged. Figure 4 shows the % of tolerated antigens and the size of the effective repertoire for networks of varying average degree for these two heterogeneous PNs, and also shows the results from the homogeneous PNs described above for reference. The results are averaged over 50 runs using the same parameters as above. Some interesting points are worth noting:

- Both classes of scale-free network and the bipartide regular network show a greater ability to tolerate certain classes of antigens than a regular homogeneous network.
- The results obtained with scale-free topologies concur with the previous conclusions with homogeneous topologies. Bi-partide topologies, either homogeneous or extremely heterogeneous, promote the emergence of high levels of tolerance for most values of $< k >$.
- Bipartide scale-free networks give rise to high levels of tolerance over a wider range of values of $< k >$. Moreover, contrary to bipartide regular potential graphs, the heterogeneous network promotes a stable self-sustained effective network for a very broad range of parameters.
- The ability of the regular bipartide network to tolerate antigens drops by one order of magnitude when then size of the network increases.

Therefore these results appear to support our suggestion that a heterogeneous topology offers a mechanism by which antigens can become protected, supporting high levels of global tolerance in a network. Two very important physical characteristics of a network — heterogeneity and cluster coefficient — therefore play a very important role in determining the ultimate functionality of a network, in terms of specifying the network topology. Further research is now continuing in this vein to add further weight to these conclusions.

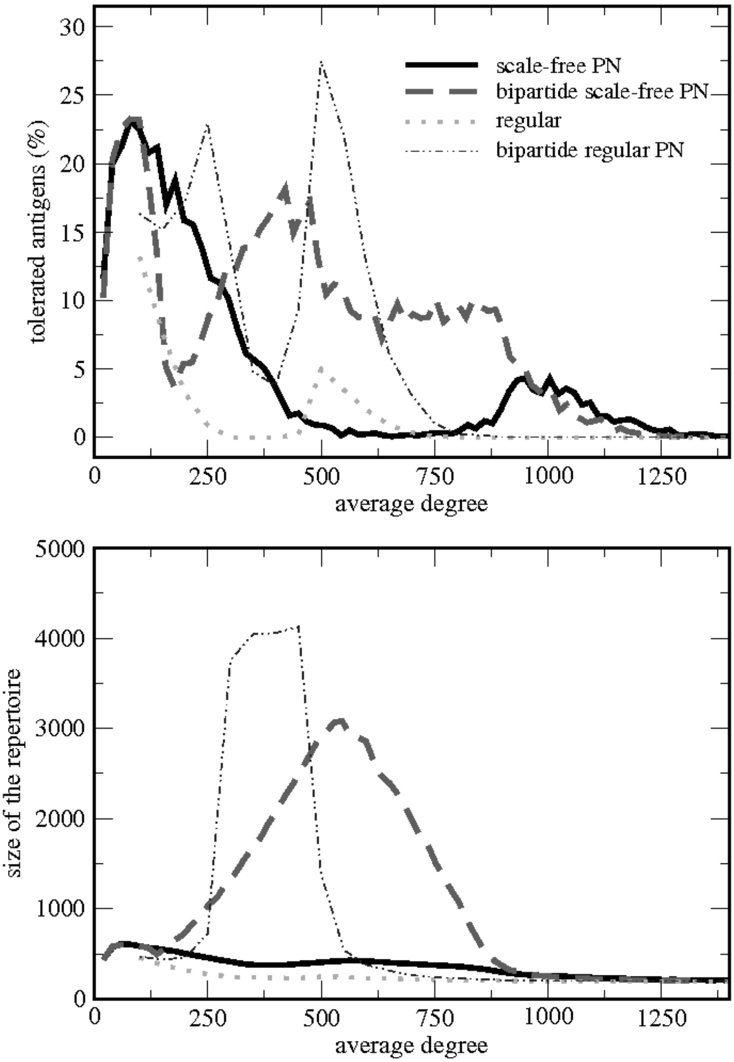


Fig. 4. The graphs show the results of simulation using a number of heterogeneous and homogeneous potential networks (both regular and bipartide in each case), as the potential average degree of the networks are varied

6 Conclusion

In this paper introduced the notion of *potential* networks as a mechanism to directly study the influence of different topologies on network behaviour. The concept facilitates study of a number of complex topologies which would be difficult to replicate through a fixed affinity measure, and allows generic hypotheses to be proposed, which are not restricted to a particular shape-space or affinity function. This is a significant departure from many previous studies. The results obtained using these networks confirm suggestions from previous work that topology plays a key role in influencing the functionality of a network but crucially, give us new information regarding the properties that a realistic potential topology (and by implication, affinity function) must exhibit. Results in the simple homogeneous scenario have shown that small differences in the underlying potential topology change completely the final outcome of the dynamical/topological system, but most importantly, we have shown that network heterogeneity can drive the topological evolution of a growing immune network, and provide protection to certain cells resulting in tolerance.

The way that nature has created the set of all possible cells, implicitly defines a given potential network. This network provide the skeleton of all dynamical processes that can evolve in a immune system scenario. Studies in other domains such as the propagation of epidemic [16,17] and evolutionary systems [19] have already shown that it is essential to study the influence of topology in networks when trying to understand their functional properties; our results suggest that this point clearly holds true for studies of immune networks as well. Moreover, as a result of their chemical and physical properties, some cells interact more than others because their physical features make such interactions extremely likely. Thus highly connected nodes arise because they have intrinsic features that allow them to connect to a higher number of partners. We can describe these nodes as *natural hubs*, i.e. nodes that were born to be hubs instead of being hubs just because of their presence in the network for a long period. These features are included in the definition of each cell and in its position in the global potential network. By disregarding the intrinsic heterogeneity of biological systems, one may lose one of the possible mechanisms that drives the topological evolution of a growing immune network. The dynamics of the concentration level of a node often produces the necessary mechanism for the tolerance and appearance of immune cells, but this in turn must be governed by the constraints imposed by the underlying potential topology.

References

1. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Rev. Mod. Phys. 74, 47–97 (2002)
2. Amaral, L.A., Scala, A., Barthelemy, M., Stanley, H.E.: Classes of small-world networks. Proc. Natl. Acad. Sci. 97(21), 11149–11152 (2000)
3. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science 286(5439), 509–512 (1999)
4. Barabási, L., Oltvai, Z.: Network biology: Understanding the cell's functional organization. Nature Reviews Genetics 5, 101–113 (2004)
5. Bersini, H.: Self-assertion vs self-recogntion: A tribute to francisco varela. In: Proceedings of ICARIS 2002 (2002)
6. Bersini, H.: Revisiting Immune Networks. In: Banzhaf, W., Ziegler, J., Christaller, T., Dittrich, P., Kim, J.T. (eds.) ECAL 2003. LNCS (LNAI), vol. 2801, pp. 191–198. Springer, Heidelberg (2003)

7. Bersini, H., Lenaerts, T., Santos, F.C.: Growing biological networks: Beyond the gene-duplication model. Journal of Theoretical Biology 241(3), 488–505 (2006)
8. 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.Theoretical Biology 169, 390–402 (1994)
9. Carneiro, J., Coutinho, A., Stewart, J.: A model of the immune network with B-T cell co-operation. II-The simulation of ontogenesis. J. Theor. Biol. 182, 531–554 (1996)
10. De Boer, R., Hoegeweg, P., Perelson, A.S.: Growth and Recruitment in an Immune Network. Theoretical and Experimental Insights into Immunology, 223–247 (1992)
11. Dorogotsev, S.N., Mendes, J.F.F.: Evolution of Networks: From Biological Nets to the Internet and WWW. Oxford University Press, Oxford (2003)
12. Guimerá, R., Amaral, L.A.N.: Functional cartography of complex metabolic networks. Nature 433(895) (2005)
13. Hart, E., Ross, P.: The impact of the shape of antibody recognition regions on the emergence of idiotypic networks. International Journal of Unconventional Computing (2005)
14. 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)
15. Hart, E., Bersini, H., Santos, F.C.: Tolerance vs intolerance: How affinity defines topology in an idiotypic network. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 109–121. Springer, Heidelberg (2006)
16. May, R.M., Lloyd, A.L.: Infection dynamics on scale-free networks. Phys. Rev. E 64(6 Pt 2), 066112 (2001)
17. Pastor-Satorras, R., Vespignani, A.: Epidemic spreading in scale-free networks. Physical Review Letters 86(3200) (2001)
18. Perelson, A.: Immune network theory. Immunological Reviews 10, 5–36 (1989)
19. Santos, F.C., Pacheco, J.M., Lenaerts, T.: Evolutionary dynamics of social dilemmas in structured heterogeneous populations. Proc. Natl. Acad. Sci. 103(9), 3490–3494 (2006)
20. Stewart, J.: The affirmation of self: a new perspective on the immune system. Artificial Life 10(3), 261–276 (2004)
21. Tanaka, R., Yi, T.M., Doyle, J.: Some protein interaction data do not exhibit power-law statistics. FEBS letters 579, 5140–5144 (2005)

A Computational Model for the Cognitive Immune System Theory Based on Learning Classifier Systems

Daniel Voigt, Henry Wirth, and Werner Dilger

Chemnitz University of Technology
D-09107 Chemnitz, Germany
{davo,henw,dilger}@informatik.tu-chemnitz.de

Abstract. In the past there have been several approaches to use Learning Classifier Systems (LCS) as a tool for modelling the functioning of the immune system. In this paper we propose a modification of the classic LCS that can be used for modelling the Cognitive Immune System Theory introduced by I. Cohen. It has been pointed out before that this alternative view of the immune system and its agents provides promising functional perspectives to the field of artificial immune systems (AIS). The characteristic features of Cohen's theory, namely degeneracy of recognition and context of immune reactions, and how they can be realized in our modified LCS are described. Moreover, we introduce the representations of the immune agents, the interactions that take place among them and the applied evolutionary mechanisms.

Keywords: Cognitive Immune System, Modelling, Learning Classifier Systems, Degeneracy, Cytokines.

1 Introduction

Most of the computational systems developed in AIS are based on the two leading theories in the field of immune system research, namely Burnet's Clonal Selection Theory [1] and Jerne's Network Theory [2]. But there have always been divergent views on immune activity – even though some of them turned out to be more biologically plausible than others (see [3]). In recent years, I. Cohen has suggested an alternative approach to understanding the functioning of the immune system as a whole which is based on the Network Theory but goes far beyond it (see [4]). He considers the immune system to be a cognitive system as it *senses* certain molecular aspects of its *environment*, creates an *internal representation* of it, and *makes decisions* about the actions that are required to keep the homeostasis of the individual. Characteristic features of his theory are the *degeneracy* of recognition events, which contrasts sharply with the assumption of monospecificity, and the emphasis on immune activity that is embedded in a *context* created by interacting immune agents. It has been pointed out that the field of AIS can benefit from computational models that are derived from such new immune theories (see [5]).

Several authors (see [6], [7], [8] and [9]) have demonstrated how LCS can be used as a framework for implementing immune-inspired computational models with

L. Nunes de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 264–275, 2007.

promising problem solving capabilities. In this paper we present a modification of the LCS and show how the important aspects of the Cognitive Immune System Theory can be modelled in this computational system. Section 2 summarizes the classic form of the LCS. In section 3 a brief introduction to the basics of the Cognitive Immune System Theory is given. Section 4 describes how the features of the theory are implemented in our modified LCS. The final section 5 contains some remarks about possible fields of application and first experiences with the implemented computational system.

2 Learning Classifier Systems

This section briefly describes the basic components and mechanisms of the machine learning paradigm that was introduced by J. Holland and is summarized under the term LCS (see [10]). There have been several variations of the underlying architectures and algorithms and so it is difficult to pick out one basic LCS standard form. In our modelling approach we focus on a form which is almost identical to the one outlined by Holland in [11]. We took this classic LCS as a starting point and modified its internal structure and processes according to the principles of Cohen's Cognitive Immune System Theory. Figure 1 shows the overall LCS and its computation loop.

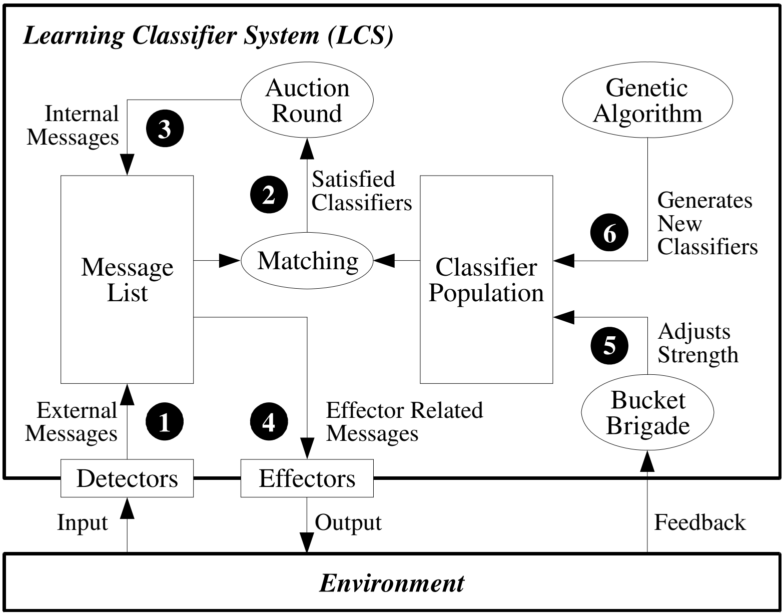


Fig. 1. The structure and internal mechanisms of Holland's classic LCS (based on [12])

A LCS consists of a population of binary rule strings called *classifiers*. A single classifier is a compound of two rule parts: the *condition* and the *action*. The condition part describes a subset of all possible *message strings* that will potentially trigger the execution of the accompanying action part. Furthermore, to each classifier a *strength* value is assigned that represents an estimate about the respective rule string's average performance in selecting appropriate actions in the past. At the beginning of the LCS loop, the rule strings of the classifier population are randomly initialized and a predefined strength value is assigned to all classifiers. Then the *message list* is cleared and the system enters the following computation loop: (1) The *detectors* add binary message strings to the message list that describe certain observations on the environment. (2) Each classifier's condition part is compared to each message located in the message list using a *matching* algorithm. All completely satisfied classifiers are added to the match set. (3) Subsequently, for each classifier of the match set a *bid* is computed that takes into account the classifier's current strength value and the specificity of its condition string. Based on these bids an *auction round* is held that stochastically determines the *winner set* by carrying out a *roulette wheel selection* among the classifiers of the match set. After that, the message list is cleared and all classifiers of the winner set publish their respective action string as a message in the message list. (4) The *effectors* read the relevant messages and perform the accompanying actions in the environment. (5) If the LCS receives feedback from the environment, the so-called *bucket brigade* algorithm adapts the strength values of certain classifiers. Thus, rule strings that triggered appropriate actions in the past are rewarded with positive feedback and become more likely to win an auction round in the future. As a result of this reinforcement strategy the LCS is able to learn from experiences in the past. (6) If necessary, weak classifiers are replaced by new ones by applying a standard *Genetic Algorithm* (GA) to the classifier population, where a rule string's strength value is interpreted as its fitness value. (7) The internal LCS loop is restarted from step 1.

3 Cognitive Immune System Theory

In this section a short introduction to Cohen's ideas regarding the cognitive immune system is given. In particular, we focus on the relevant concepts for understanding our modelling approach presented in section 4. For in-depth information on Cohen's immune theory see [4].

The immune system is an embedded biological system that maintains the body and protects it from harmful influences. These factors may come from the outside or the inside of the organism and are summarized under the term *antigens*. While operating, the immune system incorporates the states of the body tissues and thus provides an immune response that is based on *context*. This differs from the classical point of view that the main purpose of the immune system is to defend the body by discriminating between self and non-self and triggering a monospecific response (see Burnet's Clonal Selection Theory in [1]). Cohen characterizes the immune system as a cognitive system on the basis of three features: it is able to *make decisions*, it creates *internal images* of its environment, and it is able to *learn* in a *self-organized* manner.

It is comprised of a multitude of specialized *cells* and *organs*. The immune organs divide the body into distinct compartments that are flown through by the immune cells. Certain compartments of the organism are responsible for producing immune cells, others for transporting them, and again others contain the manifold interactions between the cells. On their way through the tissues the cells gather information on the tissues' state, the presence of antigens, and the activity of other immune cells. This information is available through molecular shapes that are presented by all antigens and tissue cells. All immune cells bear certain *receptor* molecules on their surfaces that enable them to bind other molecules complementarily. These recognized molecules are summarized as *ligands*. An important statement of Cohen's theory is that a receptor is able to bind *more than one* single ligand – a property referred to as *degeneracy*. The term *affinity* describes the specific binding energy between receptor and ligand that arises from their degree of molecular complementarity – the higher the affinity the higher the probability of a successful binding event. In addition, the cellular recognition is also influenced by the concentration of the respective ligand. So a relatively low ligand concentration with high affinity already causes the receptor to bind, but the low affinity of a ligand can be compensated by a high concentration as well.

After completing this distributed recognition phase in the body tissues, the immune cells gather in the lymph nodes and exchange information by producing interaction molecules called *cytokines*. These molecules provide the *context* for the immune cells to react according to the state of the organism. So on the one hand, the cells react to observations in the tissues by producing certain cytokines, and on the other hand, they react to the reactions of other immune cells by recognizing their produced cytokines. Cohen terms this special interaction between immune cells *co-respondence*.

As a result of this second-order decision-making process an apt immune response is elicited. Specialized *effector cells* move back to the body tissues and execute the made immune decisions. So despite the degenerate perceptions of its individual cells, the immune system is able to produce a specific response as a whole. The three phases of immune cell activity (recognition, decision-making, execution) follow each other continuously and thus can be described as the *functional loop* of the immune system. In each phase spatial *proximity* is an important prerequisite for the interactions between immune cells and antigens and between immune cells alone.

By means of several *evolutionary mechanisms* the more or less randomly generated population of immune cells is structured by sorting out inefficient elements. Only those immune cells are selected and included in the repertoire that show an adequate level of recognition and response in regard to certain molecular inputs. The cells that undermine the immune system's body maintaining function are deleted. As a result of this *self-organized* adaptation process an immunocompetent cell repertoire emerges and the immune system *learns* to handle the antigenic influences efficiently.

4 Our Modelling Approach

In the following section, the two systems whose basic elements and mechanisms have been introduced before are brought together: we take Holland's classic LCS as a starting point for outlining a computational model of Cohen's immune theory. Due to the fact that some of the characteristic features of Cohen's theory cannot be adequately modelled in a classic LCS, we modify the LCS with regard to Cohen's immunological principles. We term this version of a LCS a *Cognitive Immune System* (CIS).

At first, the necessary concepts and representations that are used in the CIS model are introduced. The interactions between the CIS elements and the resulting dynamics of the computational system are presented afterwards. Then, we describe the implementation of the evolutionary mechanisms that are used for adapting the system's behaviour. Finally, the whole CIS is summarized by presenting its full computation loop and comparing its constituent parts to the classic LCS form.

4.1 Representations

Firstly, we assume that the classifier population of the LCS is the counterpart to the immune cell repertoire. So an immune cell of the CIS is equivalent to a single classifier rule string. We introduce two necessary symbol sets: the set C that contains the *cytokines* $\{c_1, ..., c_l\}$ where $l \in \square^{+}$, and the set A that contains the *actions* $\{a_1, ..., a_m\}$ where $m \in \square^{+}$. The former describes the cytokines that potentially can be produced and sensed in the CIS, and the latter describes the actions that can be performed by the immune cells. Both sets are determined by the user of the system before starting and remain unchanged during the whole computation loop. It is for the user to decide which internal coding mechanism is applied to the elements of the sets – so for example, symbolic or numerical representations can be used here.

In order to model the signal processing chain of the CIS according to the principles of Cohen's theory, we modify the architecture of a classic LCS as follows: the message list is split into three distinct parts. So we obtain an *antigen message list*, a *cytokine message list* and an *action message list*. The main inputs of the immune system consist of *antigens* and *cytokines* that are produced by tissue cells. These molecules can be found in the CIS in form of *messages*. An antigen is represented by a string from the alphabet {0, 1} that has the length $k \in \square^{+}$ – although other representations can be used here (e.g. symbolic). Such a binary antigen string can be interpreted as an abstract description of a molecular shape. It is placed by the detectors in the antigen message list during the computation loop. A tissue-produced cytokine is represented by an element of the set C, which has already been defined above. The internal representation of a cytokine is published by the detectors into the cytokine message list. So the two kinds of messages (antigens and tissue-produced cytokines) can be seen as counterparts to the external messages of the classic LCS. The outputs of the CIS consist of elements of the set A, that also has been defined above. These abstract descriptions of actions are published in the action message list.

In our modelling approach the different classes of immune cells are integrated into one hybrid immune cell type. A single CIS immune cell is made of five parts: an *antigen receptor* string $R_{Antigen} \in \{0, 1\}^k$, where $k \in \square^{+}$, that describes the cell's possible perceptions; a *positive cytokine receptor* $R_{Cytokine+} \in C$, that describes the type of cytokine that stimulates the cell; a *negative cytokine receptor* $R_{Cytokine-} \in C$, that describes the type of cytokine that inhibits the cell; a *cytokine response message* $M_{Cytokine} \in C$, that is published by the cell in case of activation; and an *action message* $M_{Action} \in A$, that is suggested by the cell for execution. In addition to the receptor and the response parts, each CIS immune cell is assigned a *lifetime* $L \in \square$, that gives an estimate about the respective cell's remaining lifetime in the repertoire.

4.2 Interactions

After the CIS detectors have placed the descriptions of the antigens and the tissue-produced cytokines in the corresponding message lists, a *matching procedure* between the elements of the immune repertoire and the available immune signals takes place. In correspondence with the matching mechanism of the LCS, the elements' mutual interaction potential is identified by comparing each element of the CIS population to each element of the antigen message list and to the elements of the cytokine message list. But the CIS matching procedure differs in important aspects from the one of the LCS which only considers completely satisfied rule strings and makes use of special wildcard symbols. Since this matching criterion is much too strict to meet with the degenerate antigenic perception of Cohen's theory, we suggest a matching mechanism that is based on the concept of a binary Hamming shape-space (see [13]). For computing the *antigen affinity* $A_{Antigen}$ between an available antigen message $M_{Antigen}$ and the antigen receptor $R_{Antigen}$ of an arbitrary immune cell, we use:

$$A_{Antigen} = \frac{1}{k}\sum_{i=1}^{k} \alpha^i, \quad \text{where } \alpha^i = \begin{cases} 1 & \text{if } M^i_{Antigen} \neq R^i_{Antigen} \\ 0 & \text{else} \end{cases} \tag{1}$$

($M^i_{Antigen}$ and $R^i_{Antigen}$ refer to the i^{th} bit of the respective string)

where k is the total number of bits in an antigen message or antigen receptor string. The sum yields the number of bits where the antigen message and the corresponding antigen receptor differ, and thus describes the strings' ability to match complementarily. The antigen affinity is subsequently obtained by normalizing the value to the interval [0, 1]. Since only a pair of identical binary strings has no affinity at all, there is a certain degenerate interaction potential between almost all elements of the immune repertoire and the antigen message list.

Furthermore, the matching mechanism yields the *cytokine affinity* $A_{Cytokine}$ between all available cytokine messages and a single cytokine receptor $R_{Cytokine}$ (either positive or negative) as follows:

$$A_{Cytokine} = \frac{1}{n}\sum_{j=1}^{n} \beta^j, \quad \text{where } \beta^j = \begin{cases} 1 & \text{if } M^j_{Cytokine} = R_{Cytokine} \\ 0 & \text{else} \end{cases} \tag{2}$$

($M^j_{Cytokine}$ refers to the j^{th} cytokine message of the list)

where n is the total amount of available cytokines in the cytokine message list. The matching procedure only takes into account whether a successful recognition event on part of the immune cell receptor takes place or not. Instead of testing the degenerate matching of complementary molecular shapes, as in the case of the antigen affinity, the cytokine matching procedure focusses on the recognition event itself. Hence, each of the two cytokine receptors of an immune cell matches only one cytokine message.

The *overall affinity* of an immune cell in regard to an actual immune situation is the result of integrating three partial affinities: firstly, the antigen affinity is computed for the immune cell's antigen receptor (using Eq. (1)), and secondly, the cytokine affinities are computed for the immune cell's positive and negative cytokine receptors respectively (using Eq. (2)). In case of an occurring empty message list, the corresponding partial affinity value is set to zero. On account of the degenerate matching procedure between antigens and antigen receptors, a selection from all possible antigen-receptor-pairs has to be made to determine the set of potentially active immune cells. Since the antigen concentration also influences the interaction potential of the pair, we suggest the use of an *affinity-based roulette wheel selection* like in the auction round of the classic LCS. By means of this algorithm a single antigen message string from the antigen message list is selected. This antigen message is taken as the basis for computing the overall affinity $A_{Overall}$ according to:

$$A_{Overall} = f(A_{Antigen} + A_{Cytokine+} - A_{Cytokine-}) \tag{3}$$

$$\text{where} \quad f(x) = \begin{cases} \frac{x}{2} & \text{if} \quad x > 0 \\ 0 & \text{else} \end{cases}$$

Thus, the term $A_{Antigen}$ describes a basic activity with regard to an antigen, that is stimulated or inhibited by the following terms $A_{Cytokine+}$ and $A_{Cytokine-}$ respectively. The function *f* maps $A_{Overall}$ to the interval [0, 1]. The overall affinity is computed for each cell of the immune repertoire. The resulting values can be interpreted as the *potential activation* of the immune cells in regard to the current immune situation that is represented by the elements of the message lists. In order to determine from the set of all immune cells the subset that actually becomes activated, in accordance to the LCS an *immune auction round* is held. This restriction of potential cell activity can be seen as a computational equivalent to the spatial distribution of the immune cells in the body. According to this, an immune cell only gets activated if it has the necessary proximity to the target molecule (under the assumption that there is sufficient affinity). Therefore, the auction round can be interpreted as the CIS counterpart to this anatomical restraint. In this selection process the immune cells place their respective overall affinity $A_{Overall}$ as their bids. Subsequently, the set of auction winners – and thus the set of active immune cells – is obtained by turning the virtual roulette wheel. In addition to this, the lifetime of the immune cells that remain inactive in this round is reduced by a certain predefined amount. If the value of an immune cell's remaining lifetime falls below a certain threshold, it is removed from the repertoire. The purpose of this automatic reduction of lifetime is to increase the selection pressure in the immune cell population (see section 4.3).

As a result of their activation the immune cells produce co-respondence signals. These intercellular signals are modelled in the CIS as internal cytokine messages that are published by the active immune cells after completing the auction round. Hence, an intercellular cytokine message is the CIS counterpart to an internal LCS message. After the cytokine message list has been cleared, all activated immune cells publish their accompanying cytokine response messages in the cytokine message list. Thus, the immune cells jointly create a new cytokine pattern. In the course of the next CIS loop, this existing pattern is supplemented by the tissue-produced cytokines which are sensed through the system's detectors. Therefore, an updated cytokine context is created that is able to influence future CIS processing cycles. By means of the cytokine message list the immune cells exchange information among each other.

Besides that, the co-respondence process serves as an internal feedback mechanism within the immune repertoire. Through their two cytokine receptors the immune cells are able to receive positive and negative feedback at the same time. Therefore, a high affinity value alone is not a guarantee for leading to an immune cell's activation. In case of a simultaneously occurring inhibiting cytokine message in the cytokine message list, the overall affinity of the immune cell would be decreased by a certain amount (depending on the cytokine's concentration) and the activation probability of the immune cell would be reduced as well. The same applies to the situation where certain cytokines can compensate an immune cell's lack of antigen affinity and consequently cause an increase of the cell's activation probability. Even the presence of antigens is not a requirement – the activation of an immune cell can already be achieved by the mere influence of a sufficient amount of stimulating cytokines. Generally, in the CIS the immune cells' antigenic perceptions arise from the context of currently available cytokine messages.

According to Cohen, the co-respondence process is a joint effort between all active immune cells. As a result it yields an immune decision that influences the effector cells' subsequent actions. In the CIS this is modelled by the active immune cells that publish their respective action message in the corresponding message list and thus suggest immune actions. So the *support* for an arbitrary action is defined as the proportion of messages in the action message list that actually suggest this specific action as the next one to be performed by the effectors. In order to determine the next performed action, a *support-based roulette wheel selection* takes place. The action message that is selected by this mechanism is passed to the effectors of the CIS for execution in the tissues.

4.3 Evolutionary Mechanisms

In order to implement the evolutionary mechanisms of Cohen's immune theory, we complement the interaction procedures with an algorithmic component that models the bone marrow's function as the immune organ where immature immune cells are produced. The constituent parts of a new immune cell are produced as follows: as an analogy to the immune cells' ability to manufacture antigen receptors somatically, the CIS antigen receptor string is created randomly from the symbols of the binary alphabet; because of the genetic restrictions of the cytokine repertoire, the symbols that are assigned to the immune cell's two cytokine receptors and the respective cytokine response message are derived from the predefined set C (see section 4.1); accordingly, the action message is derived from the set A; the immune cell's lifetime value is set to a predefined constant.

In correspondence to the selection step that takes place in the thymus, all immature immune cells are subject to a testing mechanism that determines whether these cells are added to the immune repertoire or not. For this test we introduce the user-defined set S of *self messages* $\{s_1, ..., s_o\}$ where $s \in \{0, 1\}^k$ and $k, o \in \mathbb{N}^+$. These self messages describe a subset of the potential messages that is regarded to be part of the body's self, and thus can be interpreted as a filter that prevents certain new immune cells from being inserted into the repertoire. The immunocompetence of an immature immune cell is determined by comparing the antigen receptor of the cell to each self message and computing their mutual affinity (using Eq. (1)). According to Cohen's immune theory, an immune cell is rejected if its antigen affinity is too high or too low. As a result of this selection step, only those cells survive that show a moderate affinity to the set of self messages.

The next selection step of the immune system is implemented by means of the CIS auction round that determines the immune cells that actually become activated. If the inhibiting effects of the cytokine messages are left aside for a moment, this selection mechanism particularly favours the immune cells that have a high antigen affinity. So the activated immune cells have the chance to prove their usefulness regarding the present immune situation and to increase their respective lifetimes by suggesting appropriate actions (see below). The inactive immune cells do not have this opportunity. Because of their low antigen affinity they cannot compete with the high affinity immune cells and as a result are displaced from the repertoire. This selection process is boosted by automatically decreasing the lifetime of inactive immune cells.

As an analogy to the affinity maturation, a subset of the activated immune cells is reproduced. This is realized by copying the constituent parts of an immune cell and thus obtaining a set of identical cell clones. While the daughter cells' respective antigen receptor string is mutated by inverting random position bits, the other parts of the immune cells remain unchanged. So the result of this cellular modification process is a set of immune cells that only differ in their antigen receptor string. Subsequently, the antigen affinities between the mutated daughter cells and the antigen message that caused their mother's activation are computed (according to Eq. (1)). By means of an affinity-based roulette wheel selection a single immune cell is selected from the set of the mother and the daughter cells. This immune cell replaces the mother cell in the immune repertoire; all other cell clones are rejected. As a result of this selection step the immune cells' antigenic perceptions may be improved.

The last selection step (which Cohen describes only allusively) is realized in the CIS through the workings of the classic LCS bucket brigade algorithm (for details see [11]). By means of the positive or negative feedback that is received from the tissues as a result of the performed immune actions, the bucket brigade adjusts the lifetime values of the cells in the immune repertoire. An immune cell that is able to gather sufficient amounts of lifetime can be seen as the CIS equivalent to a *memory cell.*

4.4 Summary

The functional components of the resulting CIS and its computation loop are summarized in Fig. 2. The first step of the CIS is to clear all three message lists and to produce an initial population of immune cells by means of the bone marrow and the thymus components. Apart from certain structural modifications, the sequence of the internal CIS computation steps is similar to the classic LCS: (1) The detectors place messages in the corresponding message lists that describe the observed state of the tissues. (2) Each element of the immune cell repertoire is compared to the available messages of the antigen and the cytokine message list. The respective affinities are computed. (3) These potential activities are taken as a basis for an auction round that selects the immune cells that actually become activated in this round. (4) An affinity maturation mechanism is applied to a certain subset of the active immune cells. In this evolutionary step the immune cells' antigenic perception may be improved by cell cloning, receptor mutation and affinity-based selection. (5) Furthermore, all activated immune cells publish their respective cytokine and action messages in the corresponding message lists.

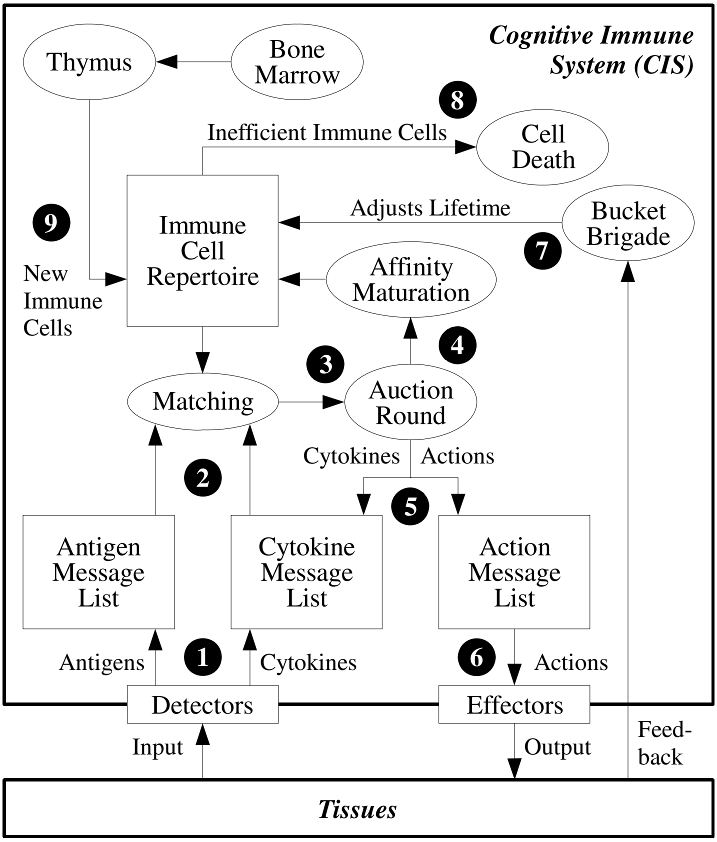


Fig. 2. The structure and internal mechanisms of the resulting CIS

(6) If there is sufficient action support, the effectors perform the suggested immune actions in the body tissues. On the one hand, from these executed actions arise new CIS inputs (see step 1), and on the other hand, the tissues react by delivering reinforcement information to the system. (7) In order to improve the overall behaviour of the CIS this feedback is used by the bucket brigade algorithm to adapt the lifetime values of the cells in the immune repertoire. (8) As a result of the several evolutionary steps, certain inefficient immune cells are removed from the repertoire. (9) New immune cells are added to the repertoire and the computation loop continues at step 1.

A comparison between the resulting CIS and the classic LCS shows, that certain LCS concepts have been included almost unchanged (e.g. message list and bucket brigade algorithm), while others have been partially reworked (e.g. rule syntax and

Table 1. Corresponding parts of the classic LCS and the CIS

Learning Classifier System (LCS)	***Cognitive Immune System (CIS)***
environment	tissues (input/output space)
message list	respective message lists for antigens, cytokines and actions
external message	antigen message, tissue-produced cytokine message
internal message	cell-produced cytokine message
classifier population	immune cell repertoire
classifier rule string	immune cell (modified syntax)
classifier condition part	antigen receptor, cytokine receptors
classifier action part	cytokine and immune action response
classifier strength	affinity (in auction round), lifetime (in bucket brigade)
rule chaining	co-respondence
GA	bone marrow, thymus, activation, lifetime reduction, affinity maturation

matching procedure), and again others have been completely replaced by immune-inspired mechanisms (e.g. the GA). A selection of the corresponding parts of the two computational systems is shown in table 1.

5 Conclusion

In this paper, we have presented a modification of a classic LCS that can be used as a computational model of Cohen's Cognitive Immune System Theory. In order to lay the foundations for our modelling approach, a short introduction to the basic elements and mechanisms of the LCS has been given. Then, we have briefly summarized the characteristic features of Cohen's immune theory. After that, the representations of the immune agents, their mutual interactions and the evolutionary mechanisms that they are subject to have been presented. The resulting computational system CIS has been compared to the parts of a classic LCS.

As in the case of the classic LCS, the possible field of application for the CIS can be seen in the domain of machine learning and problem solving. In particular, the CIS can be used for all tasks that involve context-based processing of signals (e.g. recognition of noisy patterns). Because of its simple internal representations and mechanisms the system can be easily adapted to a wide variety of computational problems.

The CIS has been implemented and several test runs with benchmark data sets from the LCS domain have been conducted. The preliminary results of these simple learning tasks are promising. It could be shown that the CIS is able to learn patterns from a presented training set by adapting its immune cell repertoire to the underlying structures of the data. But more complex tests regarding the computational system's learning capabilities remain to be done.

References

1. Burnet, F.M.: The Clonal Selection Theory of Acquired Immunity. Cambridge University Press, Cambridge (1959)
2. Jerne, N.K.: Towards a Network Theory of the Immune System. Ann. Immunol. (Inst. Pasteur) 125C, 373–389 (1974)
3. Bibel, D.J.: Milestones in Immunology - A Historical Exploration. Science Tech Publ. Madison, Wis. (1988)
4. Cohen, I.R.: Tending Adam's Garden: Evolving the Cognitive Immune Self. Elsevier Academic Press, Amsterdam (2000)
5. 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)
6. Farmer, J.D., Packard, N.H., Perelson, A.S.: The Immune System, Adaptation & Learning. Physica D 22, 187–204 (1986)
7. Hofmeyr, S.A., Forrest, S.: Architecture for an Artificial Immune System. Evolutionary Computation 8 (2000)
8. Vargas, P.A., de Castro, L.N., Michelan, R., Von Zuben, F.J.: 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)
9. Vargas, P.A., de Castro, L.N., Von Zuben, F.J.: Mapping Artificial Immune Systems into Learning Classifier Systems. In: Lanzi, P.L., Stolzmann, W., Wilson, S.W. (eds.) Learning Classifier Systems. LNCS (LNAI), vol. 2661, pp. 163–186. Springer, Heidelberg (2003)
10. Holland, J.H., Reitman, J.S.: Cognitive Systems Based on Adaptive Algorithms. In: Waterman, D.A., Hayes-Roth, F. (eds.): Pattern-directed inference systems. Academic Press, New York (1978) (reprinted). In: Fogel, D.B. (ed.) Evolutionary Computation. The Fossil Record. IEEE Press, New York (1998)
11. Holland, J.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, 1975. Republished by the MIT Press (1992)
12. Nissen, V.: Einführung in Evolutionäre Algorithmen. Vieweg, Braunschweig (1997)
13. De Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, London (2002)

Motif Detection Inspired by Immune Memory

William Wilson, Phil Birkin, and Uwe Aickelin

School of Computer Science, University of Nottingham, UK
{wow,pab,uxa}@cs.nott.ac.uk

Abstract. The search for patterns or motifs in data represents an area of key interest to many researchers. In this paper we present the Motif Tracking Algorithm, a novel immune inspired pattern identification tool that is able to identify variable length unknown motifs which repeat within time series data. The algorithm searches from a completely neutral perspective that is independent of the data being analysed and the underlying motifs. In this paper we test the flexibility of the motif tracking algorithm by applying it to the search for patterns in two industrial data sets. The algorithm is able to identify a population of motifs successfully in both cases, and the value of these motifs is discussed.

1 Introduction

The investigation and analysis of time series data is a popular and well studied area of research. Common goals of time series analysis include the desire to identify known patterns in a time series, to predict future trends given historical information and the ability to classify data into similar clusters. These processes generate summarised representations of large data sets that can be more easily interpreted by the user.

Historically, statistical techniques have been applied to this problem domain. However, the use of Immune System inspired (IS) techniques in this field has remained fairly limited. In our previous work [15] we proposed an IS approach to identify patterns embedded in price data using a population of trackers that evolve using proliferation and mutation. This early research proved successful on small data sets but suffered when scaled to larger data sets with more complex motifs. In this paper we describe the Motif Tracking Algorithm (MTA), a deterministic but non-exhaustive approach to identifying repeating patterns in time series data, that directly addresses this scalability issue.

The MTA represents a novel Artificial Immune System (AIS) using principles abstracted from the human immune system, in particular the immune memory theory of Eric Bell [16]. Implementing principles from immune memory to be used as part of a solution mechanism is of great interest to the immune system community and here we are able to take advantage of such a system. The MTA implements the Bell immune memory theory by proliferating and mutating a population of solution candidates using a derivative of the clonal selection algorithm [3].

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 276–287, 2007.

A subsequence of a time series that is seen to repeat within that time series is defined as a motif. The objective of the MTA is to find those motifs. The power of the MTA comes from the fact that it has no prior knowledge of the time series to be examined or what motifs exist. It searches in a fast and efficient manner and the flexibility incorporated in its generic approach allows the MTA to be applied across a diverse range of problems.

Considerable research has already been performed on identifying *known* patterns in time series [9]. In contrast little research has been performed on looking for *unknown* motifs in time series. This provides an ideal opportunity for an AIS driven approach to tackle the problem of motif detection, as a distinguishing feature of the MTA is its ability to identify *variable length unknown* patterns that repeat in a time series using an evolutionary system. In many data sets there is no prior knowledge of what patterns exist so traditional detection techniques are unsuitable. In this paper we test the generic properties of the MTA by applying it to motif identification in two industrial data sets to asses its ability to find variable length unknown motifs.

The paper is structured as follows, Section 2 provides a discussion of the work that has been performed in motif detection, then various terms and definitions used by the MTA are introduced in Section 3. The pseudo code for the MTA is described in Section 4. Section 5 presents the results of the MTA when applied to the two industrial data sets before moving on to conclude in Section 6.

2 Related Work

The search for patterns in data is relevant to a diverse range of fields, including biology, business, finance, and statistics. Work by Guan [6] addresses DNA pattern matching using lookup table techniques that exhaustively search the data set to find recurring patterns. Investigations using a piecewise linear segmentation scheme [7] and discrete Fourier transforms [4] provide examples of mechanisms to search a time series for a particular motif of interest. Work by Singh [12] searches for patterns in financial time series by taking a sequence of the most recent data items and looks for re-occurrences of this pattern in the historical data. An underlying assumption in all these approaches is that the pattern to be found is known in advance. The matching task is therefore much simpler as the algorithm just has to find re-occurrences of that particular pattern.

The search for unknown motifs is at the heart of the work conducted by Keogh et al. Keoghs probabilistic [2] and viztree algorithms [8] are very successful in identifying unknown motifs but they require additional parameters compared to the MTA. They also assume prior knowledge of the length of the motif to be found, so the motif is “only partially unknown”. Motifs longer and potentially shorter than this predefined length may remain undetected in full. Work by Tanaka [13] attempts to address this issue by using minimum description length to discover the optimal length for the motif. Fu et al. [5] use self-organising maps to identify unknown patterns in stock market data, by representing patterns as perceptually important points. This provides an effective solution but again the patterns found are limited to a predetermined length.

A more flexible approach is seen in the TEIRESIAS algorithm [11] able to identify patterns in biological sequences. TEIRESIAS finds patterns of an arbitrary length by isolating individual building blocks that comprise the subsets of the pattern, these are then combined into larger patterns. The methodology of building up motifs by finding and combining their component parts is at the heart of the MTA. The MTA takes an IS approach evolving a population of trackers that is able to detect motifs by making fewer assumptions about the data set and the potential motifs. It focuses on the search for unknown motifs of an arbitrary length leading to a novel and unique solution.

3 Motif Detection: Terms and Definitions

Here we define some of the terms used by the MTA.

Definition 1. *Time series.* A time series T = t_1,...,t_m is a time ordered set of m real or integer valued variables. In order to identify patterns in T we break T up into subsequences of length n using a sliding window mechanism.

Definition 2. *Motif.* A subsequence from T that is seen to repeat at least once throughout T is defined as a motif. We use Euclidean distance to examine the relationship between two subsequences C_1 and C_2, $ED(C_1, C_2)$ against a match threshold r. If $ED(C_1, C_2) \leq r$ the subsequences are deemed to match and thus are saved as a motif. The motifs prevalent in a time series are detected by the MTA through the evolution of a population of trackers.

Definition 3. *Tracker.* A tracker represents a signature for a motif sequence that is seen to repeat. It has within it a sequence of 1 to w symbols that are used to represent a dimensionally reduced equivalent of a subsequence. The subsequences generated from the time series are converted into a discrete symbol string. The trackers are then used as a tool to identify which of these symbol strings represent a recurring motif. The trackers also include a match count variable to indicate the level of stimulation received during the matching process.

4 The Motif Tracking Algorithm

This Section provides a listing of the MTA pseudo code along with a description of its main operations. We direct the readers attention to [16] for a more in depth description of this algorithm, along with a review of the immunological inspiration behind the MTA, which we do not have time to cover here. The parameters required in the MTA include the length of a symbol s, the match threshold r, and the alphabet size a.

MTA Pseudo Code

```
Initiate MTA (s, r, a)
Convert Time series T to symbolic representation
Generate Symbol Matrix S
Initialise Tracker population to size a
While ( Tracker population > 0 )
{
     Generate motif candidate matrix M from S
     Match trackers to motif candidates
     Eliminate unmatched trackers
     Examine T to confirm genuine motif status
     Eliminate unsuccessful trackers
     Store motifs found
     Proliferate matched trackers
     Mutate matched trackers
}
Memory motif streamlining
```

Convert Time Series T to Symbolic Representation. The MTA takes as input a univariate time series consisting of real or integer values. Taking the first order difference of T we look at movements between data points allowing a comparison of subsequences across different amplitudes. To further minimise amplitude scaling issues we normalise the time series. In our previous work [15] the algorithm investigated motifs through consideration of each data point individually, creating a solution that was not scalable to larger data sets. In the MTA this problem is resolved as we investigate motifs by combining individual data points into sequences and comparing and combining those sequences to form motifs.

Piecewise Aggregate Approximation (PAA) [2] is used to discretise the time series. PAA is a powerful compression tool that uses a discrete, finite symbol set to generate a dimensionally reduced version of a time series that consists of symbol strings. This intuitive representation has been shown to rival more sophisticated reduction methods such as Fourier transforms and wavelets [2].

Using PAA we slide a window of size s across the time series T one point at a time. Each sliding window represents a subsequence from T. The MTA calculates the average of the values from the sliding window and uses that average to represent the subsequence. The MTA converts this average into a symbol string. The user predefines the size a of the alphabet used to represent the time series T. Given T has been normalised we can identify the breakpoints for the alphabet characters that generate a equal sized areas under the Gaussian curve [2]. The average value calculated for the sliding window is then examined against the breakpoints and converted into the appropriate symbol. This process is repeated for all sliding windows across T to generate m-s+1 subsequences, each consisting of symbol strings comprising one character.

Generate Symbol Matrix *S*. The string of symbols representing a subsequence is defined as a **word**. Each word generated from the sliding window is entered into the symbol matrix S. The MTA examines the time series T using these words and not the original data points to speed up the search process. Symbol string comparisons can be performed efficiently to filter out bad motif candidates, ensuring the computationally expensive Euclidean distance calculation is only performed on those motif candidates that are potentially genuine.

Having generated the symbol matrix S, the novelty of the MTA comes from the way in which each generation a selection of words from S, corresponding to the length of the motif under consideration, are extracted in an intuitive manner as a reduced set and presented to the tracker population for matching.

Initialise Tracker Population to Size a. The trackers are the primary tool used to identify motif candidates in the time series. A tracker comprises a sequence of 1 to w symbols. The symbol string contained within the tracker represents a sequence of symbols that are seen to repeat throughout T.

Tracker initialisation and evolution is tightly regulated to avoid proliferation of ineffective motif candidates. The initial tracker population is constructed of size a to contain one of each of the viable alphabet symbols predefined by the user. Each tracker is unique, to avoid unnecessary duplication.

Trackers are created of a length of one symbol and matched to motif candidates via the words presented from the stage matrix S. Trackers that match a word are stimulated and become candidates for proliferation as they indicate words that are repeated in T. Given a motif and a tracker that matches part of that motif, proliferation enables the tracker to extend its length by one symbol each generation until its length matches that of the motif.

Generate Motif Candidate Matrix *M* from *S*. The symbol matrix S contains a time ordered list of all words, each containing just one symbol, that are present in the time series T. Neighbouring words in S contain significant overlap as they were extracted via sliding windows. Presenting all words in S to the tracker population would result in inappropriate motifs being identified between neighbouring words. To prevent this issue such 'trivial' match candidates are removed from the symbol matrix S in a similar fashion to that used in [2].

Trivial match elimination is achieved as a word is only transferred from S for presentation to the tracker population if it differs from the previous word extracted. This allows the MTA to focus on significant variations in the time series and prevents time being wasted on the search across uninteresting variations.

Excessively aggressive trivial match elimination is prevented by limiting the maximum number of consecutive trivial match eliminations to s, the number of data points encompassed by a symbol. In this way a subsequence can eliminate as trivial all subsequences generated from sliding windows that start in locations contained within that subsequence (if they generate the same symbol string) but no others. The reduced set of words selected from S is transferred to the motif candidate matrix M and presented to the tracker population for matching.

Match Trackers to Motif Candidates. During an iteration each tracker is taken in turn and compared to the set of words in M. Matching is performed using a simple string comparison between the tracker and the word. A match occurs if the comparison function returns a value of 0, indicating a perfect match between the symbol strings. Each matching tracker is stimulated by incrementing its match counter by 1.

Eliminate Unmatched Trackers. Trackers that have a match count >1 indicate symbols that are seen to repeat throughout T and are viable motif candidates. Eliminating all trackers with a match count < 2 ensures the MTA only searches for motifs from amongst these viable candidates. Knowledge of possible motif candidates from T is carried forward by the tracker population. After elimination the match count of the surviving trackers is reset to 0.

Examine T to Confirm Genuine Motif Status. The surviving tracker population indicates which words in M represent viable motif candidates. However motif candidates with identical words may not represent a true match when looking at the time series data underlying the subsequences comprising those words. In order to confirm whether two matching words X and Y, containing the same symbol strings, correspond to a genuine motif we need to apply a distance measure to the original time series data associated with those candidates. The MTA uses the Euclidean distance to measure the relationship between two motif candidates $ED(X,Y)$ [16].

If $ED(X,Y) \leq$ r a motif has been found and the match count of that tracker is stimulated. A memory motif is created to store the symbol string associated with X and Y. The start locations of X and Y are also saved. For further information on the derivation of this matching threshold please refer to [16].

The MTA then continues its search for motifs, focusing only on those words in M that match the surviving tracker population in an attempt to find all occurrences of the potential motifs. The trackers therefore act as a pruning mechanism, reducing the potential search space to ensure the MTA only focuses on viable candidates.

Eliminate Unsuccessful Trackers. The MTA now removes any unstimulated trackers from the tracker population. These trackers represent symbol strings that were seen to repeat but upon further investigation with the underlying data were not proven to be valid motifs in T.

Store Motifs Found. The motifs identified during the confirmation stage are stored in the memory pool for review. Comparisons are made to remove any duplication. The final memory pool represents the compressed representation of the time series, containing all the re-occurring patterns found.

Proliferate Matched Trackers. Proliferation and mutation are needed to extend the length of the tracker so it can capture more of the complete motif. At the end of the first generation the surviving trackers, each consisting of a word with a single symbol, represent all the symbols that are applicable to the motifs in T. Complete motifs in T only consist of combinations of these symbols. These trackers are stored as the mutation template for use by the MTA.

Proliferation and mutation to lengthen trackers will only involve symbols from the mutation template and not the full symbol alphabet, as any other mutations would lead to unsuccessful motif candidates. During proliferation the MTA takes each surviving tracker in turn and generates a number of clones equal to the size of the mutation template. The clones adopt the same symbol string as their parent.

Mutate Matched Trackers. The clones generated from each parent are taken in turn and extended by adding a symbol taken consecutively from the mutation template. This creates a tracker population with maximal coverage of all potential motif solutions and no duplication. This process forms the equivalent of the short term memory pool identified by Bell [1] and is illustrated in more detail in [16].

The tracker pool is fed back into the MTA ready for the next generation. A new motif candidate matrix M consisting of words with two symbols must now be formulated to present to the evolved tracker population. In this way the MTA builds up the representation of a motif one symbol at a time each generation to eventually map to the full motif using feedback from the trackers.

Given the symbol length s we can generate a word consisting of two consecutive symbols by taking the symbol from matrix S at position i and that from position i+s. Repeating this across S, and applying trivial match elimination, the MTA obtains a new motif candidate matrix M in generation two, each entry of which contains a word of two symbols, each of length s.

The MTA continues to prepare and present new motif candidate matrix data to the evolving tracker population each generation. The motif candidates are built up one symbol at a time and matched to the lengthening trackers. This flexible approach enables the MTA to identify unknown motifs of a variable length. This process continues until all trackers are eliminated as non matching and the tracker population is empty. Any further extension to the tracker population will not improve their fit to any of the underlying motifs in T.

Memory Motif Streamlining. The MTA streamlines the memory pool, removing duplicates and those encapsulated within other motifs to produce a final list of motifs that forms the equivalent of the long term memory pool.

5 Results

Here we examine the MTA's performance on two publicly available industrial data sets. The MTA was written in C++ and run on a Windows XP machine with a Pentium M 1.7 Ghz processor with 1Gb of RAM.

5.1 Steamgen Data

The steamgen data set was generated using fuzzy models applied to the model of a steam generator at the Abbott Power Plant in Champaign [10] and is available from http://homes.esat.kuleuven.be/∼tokka/daisydata.html. The

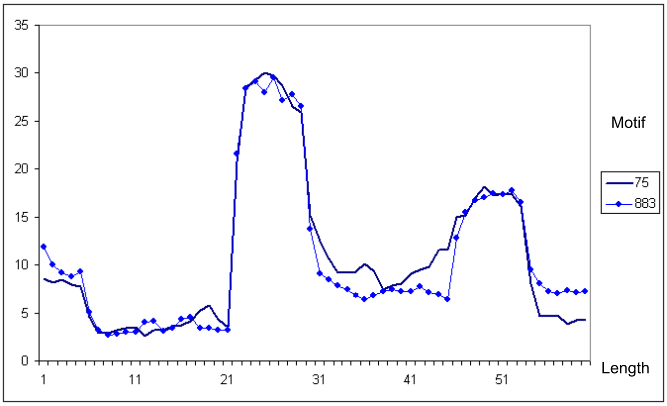


Fig. 1. The plot of a motif found in the steamgen data by the MTA. It consists of the subsequences starting at locations 75 and 883, both of length 60. The X axis refers to the motif length, whilst the Y axis refers to steam flow.

steamgen data set consists of every tenth observation taken from the steam flow output, starting with the first observation. This specific data selection was used by Keogh and has been followed for the purposes of comparison.

The steamgen data set contains 960 items with significant amplitude variation. Parameters s = *10*, a = *6*, r = *0.5* were established as suitable after numerous runs of the MTA. Sensitivity analysis on these parameters can be found in [16]. The MTA identified 104 motifs of lengths varying between ten and 60 data points. Some of the motifs of length ten are seen to repeat up to 15 times throughout the data, others of length 20 are noted to repeat up to four times. One significant motif of length 60, seen to occur twice in the data, at locations 75 and 833, dominates the motif pool. This motif is plotted in Figure 1.

In order to provide some grounding for the MTA, we compared the MTA result to that of the Keogh's probabilistic motif search algorithm [2]. Keogh was kind enough to provide a teaching version of the algorithm which we applied to the steamgen data, using parameters established by Keogh. Given a predefined length of 80, the algorithm was able to identify a dominant motif consisting of sequences starting at points 66 and 874, that is consistent with the motif found by the MTA, as illustrated in Figure 2.

Comparing Figures 1 and 2 it appears that the MTA has only detected a subset of the motif found by the probabilistic algorithm, missing off the first and last ten data points of the longer motif. However, the Euclidean distances across the first and last ten point sequences are 5.48 and 11.17 respectively. Given s = *10* and r=*0.5* per unit, a match threshold of 5.0 is applied to each ten point sequence, resulting in the rejection of both omitted subsequences as non matching. Sensitivity analysis was performed on the MTA to changes in s, r, and a [16]. The MTA is sensitive to changes in s and r but not a. Reducing

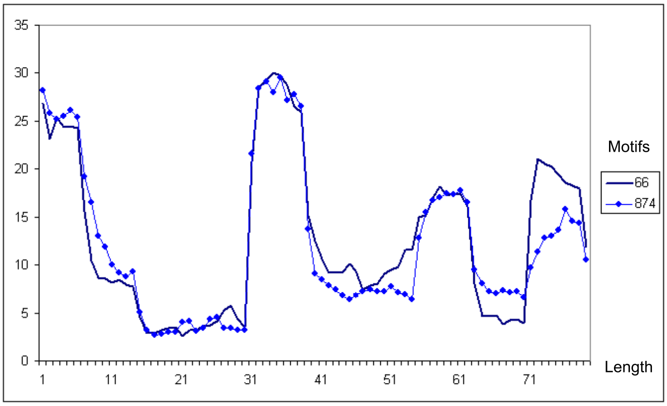


Fig. 2. The plot of a motif found in the steamgen data by Keoghs probabilistic algorithm. It consists of the subsequences starting at locations 66 and 874, both of length 80. The X axis refers to the motif length, whilst the Y axis refers to steam flow.

s from 20 to 10 and then to 5 increases execution times by 278% and 776% respectively, the more detailed search takes significantly longer. Reducing r by 50% reduces execution time by approximately 92% as the stricter bind condition reduces the number of motif candidates investigated.

If the user is aware of the motif length then Keogh's probabilistic algorithm produces satisfactory results. It could be run for alternative motifs lengths to find improved motifs, however this reduces the algorithms effectiveness. Without knowledge of motif length the MTA provides a successful alternative. It is able to dynamically build up identification of the motif, symbol by symbol, until the match threshold is exceeded. The search process is driven by the match criteria and not a predetermined motif length, leading to better fitting motifs.

5.2 Power Demand Data

Having compared the MTA to an alternative approach we now focus the MTA on the power demand data set (www.cs.ucr.edu/∼eamonn/TSDMA/index.html) which contains 35,040 fifteen minute averaged values of power demand (KW) for the ECN research centre during 1997 [14]. A subset of 5,000 data points was extracted from data point 5,000 onwards for evaluation. Figure 3 plots this data subset, the five week day peaks in power demand are clearly evident, with minimal demand seen to occur over the weekends.

Running the MTA with the previously determined alphabet size $a = 6$, and increasing $s = 500$ and $r = 4$ given the larger data set and the magnitude of actual data values, the MTA is able to identify 18 motifs within 798,298ms. The most frequently repeated motif A is plotted in Figure 3. Motif A represents the

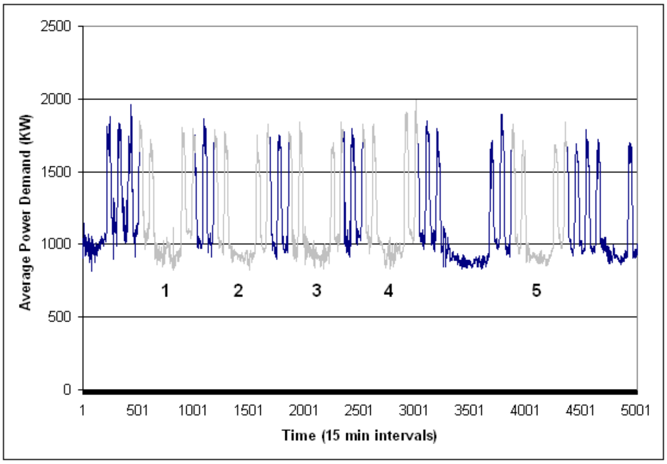


Fig. 3. Power demand data subset with Motif A of length 500 highlighted in light grey, with five occurrences (listed 1 to 5) starting at locations 508, 1182, 1854, 2525 and 3869

power demand from Thursday through to Tuesday, including a normal weekend with minimal demand on Saturday and Sunday, that is seen to occur five times. The intervals between the first four occurrences of the motif are approximately 675 data points, or seven days given that each data point is a 15 minute interval, whilst that between the fourth and fifth occurrences is 14 days. This implies a potential motif is missing from data point 3200 to 3700. However this subsequence relates to the period from the 26th to the 31st March 1997 and in the Netherlands the 28th and 31st of March are bank holidays during which there was no power demand. The sequence from 3200 to 3700 is therefore not consistent with motif A and was omitted. This simple case shows the MTA has been able to find a motif that represents all occurrences of a two day weekend that has no associated bank holidays.

Reducing the symbol size s to 400 for a more detailed search, the MTA identifies 21 motifs in 661,902ms. One Motif B, of length 400, is seen to occur twice at locations 2880 and 3648, see Figure 4. The MTA found a motif that corresponds to the two weeks that incorporate a bank holiday.

One could argue that the four day patterns found in motif B should be found to repeat as a subset of all the other five day working weeks. However the MTA is able to distinguish between the bank holiday weeks and normal five day working weeks as it identified another motif C of length 1308, seen to occur twice in the data at positions 539 and 1884. This motif encapsulated a fortnight of normal working days that was seen to repeat twice, covering the period up to the start of motif B. Motif C did not occur for a third time after this due to the existence of the bank holidays which broke the matching criteria for that sequence.

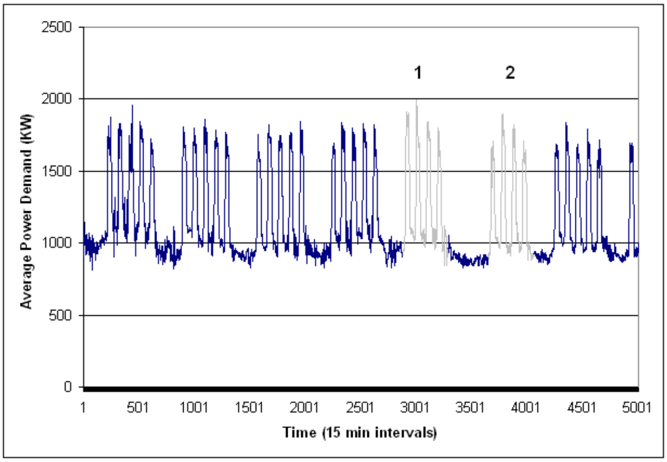


Fig. 4. Power demand data subset with Motif B of length 400 highlighted in light grey, occurring twice as labelled 1 and 2, at locations 3924 and 3648

6 Conclusion

Motifs and patterns are key tools for use in data analysis. By extracting motifs that exist in data we gain some understanding as to the nature and characteristics of that data. The motifs provide an obvious mechanism to cluster, classify and summarise the data, placing great value on these patterns. Whilst most research has focused on the search for *known* motifs, little research has been performed looking for *variable length unknown* motifs in time series. The MTA takes up this challenge, building on our earlier work to generate a novel immune inspired approach to evolve a population of trackers that seek out and match motifs present in a time series. The MTA uses a minimal number of parameters with minimal assumptions and requires no knowledge of the data examined or the underlying motifs, unlike other alternative approaches. Previous issues of scalability were addressed by using a discrete, finite symbol set to generate a dimensionally reduced version of the time series for investigation.

The MTA was evaluated using two industrial data sets and the algorithm was able to identify a motif population for each. In the steamgen data set a dominant motif was identified and compared to results from an alternative author. The ability of the MTA to find improved variable length motifs due to it's immune memory inspired tracker evolution was also highlighted, a distinguishing feature over other algorithms. In the power demand data set the MTA was able to identify motifs that had meaningful significance to the user. The MTA found as motifs i) those periods that correspond to weekends not associated with bank holidays, ii) the four day working weeks that contain a bank holiday, and

iii) the normal five day working weeks. From these results we believe the MTA offers a valuable contribution to an area of research that at present has received surprisingly little attention.

References

1. Bell, E.B., Sparshott, S.M., Bunce, C.: CD4+ T-cell memory, CD45R subsets and the persistence of antigen - a unifying concept. Immunology Today 19, 60–64 (1998)
2. Chiu, B., Keogh, E., Lonardi, S.: Probabilistic discovery of time series motifs. In: SIGKDD (August 2003)
3. 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)
4. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time series databases. In: proceedings of the SIGMOD conference, pp. 419–429 (1994)
5. Fu, T.C., Chung, F.L., Ng, V., Luk, R.: Pattern discovery from stock market time series using self organizing maps. In: Workshop notes of KDD2001 workshop on temporal data mining, San francisco, CA, pp. 27–37 (2001)
6. Guan, X., Uberbacher, E.C.: A fast look up algorithm for detecting repetitive DNA sequences. Pacific symposium on biocomputing, Hawaii, IEEE Tran. Control Systems Tech. (December 1996)
7. Keogh, E., Smyth, P.: A probabilistic approach to fast pattern matching in time series databases. In: Proceedings of the third international conference of knowledge discovery and data mining, pp. 20–24 (1997)
8. Lin, J., Keogh, E., Lonardi, S.: Visualizing and discovering non trivial patterns in large time series databases. Information visualization 4(2), 61–82 (2005)
9. Lin, J., Keogh, E., Lonardi, S., Patel, P.: Finding motifs in time series. In: The 2nd workshop on temporal data mining, at the 8th ACM SIGKDD international conference on knowledge discovery and data mining, July 2002, ACM Press, New York (2002)
10. Pellegrinetti, G., Benstman, J.: Nonlinear control oriented boiler modelling, a benchamrk problem for controller design. IEEE Tran. Control Systems Tech. 4(1) (January 1996)
11. Rigoutsos, I., Floratos, A.: Combinatorial pattern discovery in biological sequences: TEIRESIAS algorithm. Bioinformatics 14(1), 55–67 (1998)
12. Singh, S.: Pattern modelling in time series forecasting. Cybernetics and systems - an international journal 31(1) (2000)
13. Tanaka, Y., Uehara, K.: Discover motifs in multi-dimensional time series using the principal component analysis and the MDL principle. In: 3rd international conference on machine learning and data mining in pattern recognition, Leipzig, Germany, pp. 252–265 (2003)
14. van Wijk, J.J, van Selow, E.R.: Cluster and calendar based visualization of time series data. In: IEEE Symposium on information vizualisation INFOVIS'99, San Francisco, October 25-26, 1999, IEEE Computer Society Press, Los Alamitos (1999)
15. Wilson, W.O., Birkin, P., Aickelin, U.: Price trackers inspired by immune memory. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 362–375. Springer, Heidelberg (2006)
16. Wilson, W.O., Birkin, P., Aickelin, U.: The motif tracking algorithm. IEEE Transactions on Evolutionary Computation, 2007 (under review)

An Immune-Inspired Approach to Speckled Computing

Despina Davoudani, Emma Hart, and Ben Paechter

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

Abstract. *Speckled Computing* offers a radically new concept in information technology that has the potential to revolutionise the way we communicate and exchange information. Specks — minute, autonomous, semi-conductor grains that can sense and compute locally and communicate wirelessly — can be sprayed into the atmosphere, onto surfaces or onto people, and will collaborate as programmable computational networks called SpeckNets which will pave the way to the goal of truly ubiquitous computing. Such is the vision of the Speckled Computing Project — however, although the technology to build such devices is advancing at a rapid rate, the software that will enable such networks to self-organise and function lags somewhat behind. In this paper, we present a framework for a self-organising SpeckNet based on Cohen's model of the immune system. We further suggest that the application of immune inspired technologies to the rapidly growing field of pervasive computation in general, offers a distinctive niche for immune-inspired computing which cannot be filled by another other known technology to date.

1 Introduction

Advances in micro-electro-mechanical systems technology, wireless communications, and digital electronics have enabled the development of exceptionally small mechanical devices that are inexpensive, low-power and capable of sensing phenomena in the physical world [1]. Such devices can be connected together in large numbers to form wireless sensor networks (WSNs). The concept of WSNs promises a wide range of new application areas, ranging from environmental applications [19,20], military applications [22] to structural health [17]. The Speckled Computing Consortium [23] is dedicated to the realisation of a new WSN platform, which promises to create minute semiconductor grains, called *specks*, around the size of one cubic millimetre [24]. Moreover, a network of specks differs from the standard sensor network in that it does not incorporate base stations, but relies solely on many minuscule nodes, each of which has limited abilities and resources. Each minute *speck* contains its own processor, memory, and communication hardware, and can be equipped with one or more sensors to monitor environmental properties. This promises a new generation of "spray-on computers" [3], in which dense networks, *SpeckNets*, consisting of thousands of nodes can be created.

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 288–299, 2007.

Specks may be scattered or sprayed on the person or surfaces, and act as a "computational aura" [23], opening up a plethora of potential applications, perhaps using tens to thousands of specks. For example, a few tens of specks may be attached to rigid object allowing tracking of the position and orientations of articulated rigid bodies [27]. Normally passive artefacts such as furniture and appliances might have specks incorporated into their structure, thus enabling them to interact with users — Wong *et al* [25] describe a scenario in which specks integrated into a reading table might detect the removal of a book from a table and automatically turn on a reading light. On detection of a fire, a speck system might automatically drop thousands of specks from the ceiling onto the floor, where they would self-organise in order to light up pathways to the nearest exit. The technology clearly promises much, yet despite great advances in hardware and micro technology enabling such devices to be built, the software required to enable these devices to function in a useful manner lags far behind.

A SpeckNet can be considered as a complex system in which the interactions of many autonomous, spatially distributed agents result in coherent global behaviours, not explicitly planned beforehand, but resulting from the process of self-organisation. The complexity of the software required to engineer such systems goes beyond the capabilities of traditional computer science and software engineering abstractions [28]. Traditional object-oriented and component-based methodologies are insufficient to model and engineer such behaviours, requiring a paradigm shift in thinking in order to tackle this problem. Increasingly, research has turned to biological systems in order to provide inspiration, for example evolution, ant colonies and the natural immune system. Indeed, a number of bio-inspired approaches to wireless sensor networks are described in [10,11]. In this paper we describe how the mammalian immune system can be used to inspire an architecture which will allow the potential of SpeckNets to be realised.

The field of Artificial Immune Systems (AIS) is now well established, and mechanisms from the immune system have inspired applications in diverse fields such as optimisation, clustering, fault detection and security. However, Hart and Timmis [14] point out that much of the work in the field fails to capitalise on the true potential of this rich metaphor, and instead uses isolated mechanisms to solve particular tasks. They suggest that the way forward for the AIS field lies with applications which exhibit a number of key properties: they consist of multiple, heterogeneous, interacting components which are easily and naturally distributed, they are embodied, exhibit homeostasis, and are required to perform life-long learning. In this paper, we describe how the SpeckNet platform exactly embodies these features, and how one particular immunological perspective, proposed by Cohen [6], can potentially provide the foundation for an architecture which will allow a SpeckNet to both *regulate its operation* so that it remains active as long as it can (i.e. maintain itself) and *fulfil its application goals.* This is the first proposal for a software platform for a self-organising SpeckNet. More generally, we suggest that the type of domain typified by SpeckNet represents a new direction for AIS research which potentially will create a new generation of distinctive and effective AIS algorithms.

The remainder of the paper is organised as follows. Section 2 provides a description of SpeckNets, their features and limitations that justify their need for an alternative approach, other than classical computing. Then, Sect. 3 presents the immune model considered to be exploited for the case of SpeckNets, followed by Sect. 4 in which the mapping between the immune model is proposed and a realisable software framework is made explicit.

2 Physical Characteristics of SpeckNets

A *SpeckNet* is an autonomous computing machine which must achieve some specified task and simultaneously organise and maintain itself. Although the ultimate goal of the SpeckNet Consortium is to create minuscule semiconductor grains around the size of one cubic millimetre [24], the first generation of these units are aimed to have dimensions of approximately five millimetres cubed. The main components that constitute the current generation of specks are:

- a computer-on-a-chip, that combines a micro-processor and memory (FLASH and RAM),
- a radio chipset, which is compatible with the IEEE 802.15.4 standard for low rate wireless personal area networks [16],
- an antenna that allows communication ranges from a few centimetres to over a few meters,
- a power supply, such as compact rechargeable batteries, and
- a number of sensors, which vary from heat sensors to accelerometers, depending on the type of application.

Sensors carried by each speck allow data capture, while in-built processing capabilities permit specks to filter data and extract information from the environment, and LED components provide specks with a feedback mechanism. Distributed functioning in SpeckNets is enabled by incorporating communication capabilities, which allow the constituent devices to interact with each other. The lack of powerful base station units forces the network to operate in a decentralised and asynchronous fashion, by sharing tasks and processes between its autonomous units. Figure 1 shows the latest version of a speck prototype, known as ProSpeckz (Programmable Specks over Zigbee Radio), the size of which is 33×22×8mm excluding battery. In addition, Fig. 2 illustrates a system-level overview of the ProSpeckz device [24]. All levels are currently in development, but, at this stage, research primarily concentrates onto hardware and firmware tiers (SpeckNet Medium Access Control and SpeckNet Network Protocol). Future generation of specks may include optics as an extra means of wireless communication, and solar cells in order to provide a renewable power source. Another possible addition is the connection of actuators to the board (e.g. a motor) to create a mobile platform for games. However, the current project assumes the use of the ProSpeckz only.

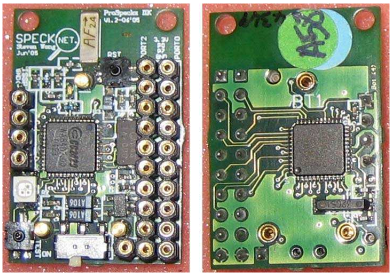


Fig. 1. The latest version (June 2005) of the speck prototype, ProSpeckz IIK (without battery), built by the SpeckNet Group; dimensions 33×22×8mm

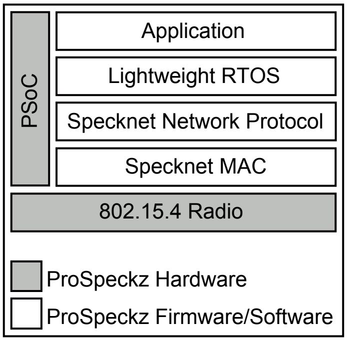


Fig. 2. ProSpeckz system-level overview [24]

Limitations of Specks. The physical characteristics of speck devices inevitably impose restrictions on the functionality of SpeckNets and of any software which might control them. First and foremost, the minute dimensions of a speck affects all constituent parts of a speck node, and the overall performance and efficiency of SpeckNets consequently. Committing to a small size physically, entails limited space for providing on board memory and power supply. Therefore, storage capacity of specks is extremely limited (about less than 10KBytes of FLASH memory and 1KBytes at most of RAM), and energy alike. Processing power is also restricted because of the above constraints, especially due to memory constraints (physical size is not an issue, as powerful 32-bit processors can be found at small dimensions of 0.5mm^2).[1]

Furthermore, wireless communication is not only unreliable but is of limited range too. To minimise the size of the antenna area (13×6mm), operation is set at high frequencies (2.4GHz), which however incurs high path losses [26]. Yet the most crucial constraint in SpeckNets is to minimise power consumption on every design level [7], which is also a common problem in many embedded sensor networks [15]. Technically, the most expensive activity in small sensor nodes is radio usage. Data transmission and reception usually draw significant amounts of power, in comparison with processor and sensors [12,24]. Sending a single bit can consume the same energy as executing 1000 instructions [18]. Energy drain leads to speck "death", which is also exacerbated by the placement in open environments, where it is not always possible to fully control operating conditions. Thus, unpredictable node failures can occur for a number of reasons, which has an unpredictable effect on the functioning of the speck network.

[1] Technical information was obtained via discussions with members of the SpeckNet group.

2.1 Mapping the Immune System to SpeckNets

The above description of specks at both the physiological and functional level suggests an immediate analogy with the natural immune system. Millions of individual cells are distributed throughout the body and communicate asynchronously using various immune molecules. They form local networks via their cellular interactions, despite the absence of some main organ acting as a coordinator. Similarly, a SpeckNet comprises hundreds of tiny devices which are spread out on surfaces or within objects, and constitute a wireless network via optical or radio frequency communication. Moreover, a SpeckNet differs from the standard sensor network in that it does not incorporate base stations but relies solely on many minute nodes, each of which has limited abilities.

Individual cells in the immune system can be considered as autonomous agents that move within the body and can exit the system at any time (by cell death due to apoptosis or necrosis). Specks are similar to immune cells, in the sense that they also are structurally basic and autonomous, since they have built-in renewable energy source. In the case of both specks and immune cells, the individual resources of any agent are modest and thus are considered "weak" and unreliable on their own. Immune cells are capable of migrating through the body; in this respect, they differ from specks which in many of the suggested applications are essentially static, the specks having no means of self-propulsion. However, just as immune cells are carried by the blood, one can envisage specks being placed in flowing liquids (for example in pipelines in order to detect and plug leaks). Mobile versions of specks have been suggested to provide a generic technology for mobile toys and robotics [24].

On a functional level, the immune system displays interesting computational properties [9]. It has developed mechanisms of learning and memory that allow the system to adapt to changes in its environment. It organises its own function and self-regulates its responses. In spite of partial cell failures and confrontations with unknown events, it still manages to cope with problematic situations and ensures robust operation over a long period of time.

Likewise, SpeckNet is envisaged to be a functional network, even though it is composed of individually feeble units. It is expected to be tolerant to perturbations that appear within the system (e.g. single specks' failure), and deal at the same time with changes in its surroundings. Self-regulation of its resources (power, processing, memory) is vital, along with the ability to operate without any external reference point of control. Finally, as mentioned in the introduction, speckled applications may vary greatly in the number of specks used to form a SpeckNet, therefore it is crucial that any network behaviour is scalable.

3 An Immune-Inspired Architecture for SpeckNets

The previous section suggests an obvious mapping between both the components and functionality of the immune system and a SpeckNet. Yet, unfortunately for the computer scientist, immunology offers a number of disparate and often conflicting theories as to how these components interact in order to achieve the desired functionality. Several “schools of thought” appear in the literature of AIS [13,8], ranging from the adoption of single immunobiological mechanisms (e.g. negative selection, clonal selection principle), to more holistic views such as the immune network hypothesis and danger theory. One view that has not explicitly inspired any computational practice thus far but has become popular in immunological circles in recent times, is the cognitive perception of the immune system first proposed by Cohen in [4,5] and popularised in his book [6]. It has, however, triggered discussion about the potentiality of the principles it offers, to be exploited in computational systems, for example by [2], who envision the next generation of AIS to draw inspiration from Cohen’s model. We present a brief summary of the relevant concepts from this theory in relation to SpeckNet — the reader is referred to [6] for a fuller description.

Cohen was one of the first to describe the immune system as *cognitive*, a property which is of great interest to SpeckNet development, as such applications are required to be fully cognisant. Cognition is usually thought of as a conscious process performed by the brain, but Cohen defines three elements that, when integrated, make up a cognitive system without consciousness. The first element is the ability of the system to exercise *options*, to make *decisions* based on a number of choices. The system contains *internal images* of its environment, which are updated based on its experience, gained via interactions or *self-organisation.* Update implies increase of information which is driven through inputs of energy and information by the world, and similar outputs generated by the system. Eventually, these interactions result in choices, and therefore the emergence of cognition.

Cohen describes three important mechanisms which contribute to the emergence of cognition; *co-respondence, pleiotropia* and *redundancy.* The concept of *co-respondence* suggests that in order to fulfil its role (maintenance, protection), the immune systemn maintains different classes of immune cells (macrophages, T cells, B cells). These cells individually see different aspects (context, P-MHC complexes, conformation) of any object that may be of immune interest, from within the body (tissues) or external (antigens). Each class of immune cell informs other immune cells about what it has seen, by expressing co-response signals (cytokines, processed peptides, interaction molecules, antibodies). The effect of these signals is that each cell modifies its own response based on the feedback it receives from the other cells [6] — essentially, although it is impossible for a cell to perceive what another cell perceives of its environment, it can perceive how another cells responds, and therefore respond to this response. This is a key point for specks; physical limitations of specks in terms of memory and limited communication abilities render it impossible for specks to communicate their perceptions from the environment to other specks in the vicinity, therefore they need to rely on minimal message passing in order to achieve a global response to environmental signals.

Pleiotropia denotes the capacity of a single immune component to produce several diverse effects. Depending on existing conditions, immune agents elicit different responses and do different, sometimes contradictory, things. For example, in the natural immune system, a T cell can kill a target cell and stimulate the growth of another. In a SpeckNet, a speck may broadcast a message indicating to others in its locality to “enter sleep mode” in order to conserve energy. However, other specks may individually decide to accept or reject this prompting, based on the importance of the calculations they are processing, at the moment when receiving the signal. Thus the sleep signal is pleiotropic, depending on the context of its reception — in case of highly important processing, the speck will turn off all of its parts but the processor, otherwise the entire speck will enter sleep mode.

Redundancy is distinguished between simple and degenerate. Simple redundancy designates the existence of multiple copies of the same element (e.g. numerous antibodies are produced during an immune response), while degenerate redundancy describes the situation in which many different immune components perform the same action. Both types are relevant to a speck network. For example, due to the extreme unreliability of individual specks, multiple copies of the same *species*[2] of speck will inevitably exist in any application, and in addition, different species of speck may respond in a similar manner to certain environmental conditions.

4 Framework

In this section, we present an initial framework based on the Cohen perspective of the immune system which will provide the basis for experimentation into an immune-inspired SpeckNet architecture. Figures 3 and 4 provide a mapping between the Cohen and SpeckNet models, which is discussed in this section.

The overall role of the immune system is to provide *maintenance* and *protection* to a host. Likewise, the role of the underlying SpeckNet in any application is to *regulate its operation*, so that it remains active as long as it can, and *fulfil its application goals*, namely to capture any interesting stimuli expressed by the environment and generate the appropriate response. As in the immune system, specks perform one or more of three basic functions; they can *sense* information from their environment; they *process* information; and they *communicate* information, i.e. carry signals. In both the immune and SpeckNet environments, cells can be *active* or *resting* (corresponding to an idle state in a speck). In addition, specks can be in a further state which has no direct immunological analogy, of *sleeping* in which a particular component is turned off and therefore does not interact with the system.

In the immune system, processes occurring in the tissues of a host provide a context for signals received by the immune system, for instance indicating infection or damage. This can be seen as analogous to the *internal* state of a speck, corresponding to the current state of (for example) its battery, malfunction or failure of any of its components and its relative location to other specks (which can be calculated by existing algorithms, for example [21]). However, specks exist

[2] Where a species refers to a set of specks containing identical sensors and processing capabilities.

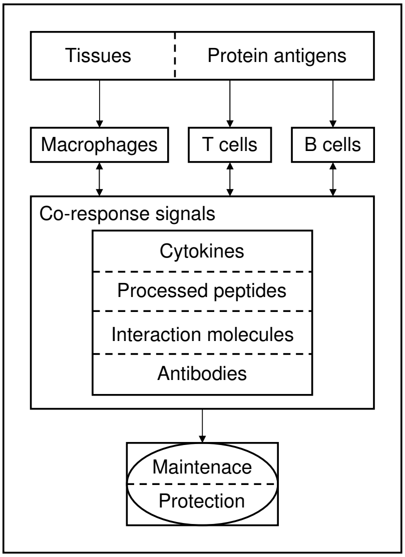


Fig. 3. Co-respondence, adapted from Cohen [6]

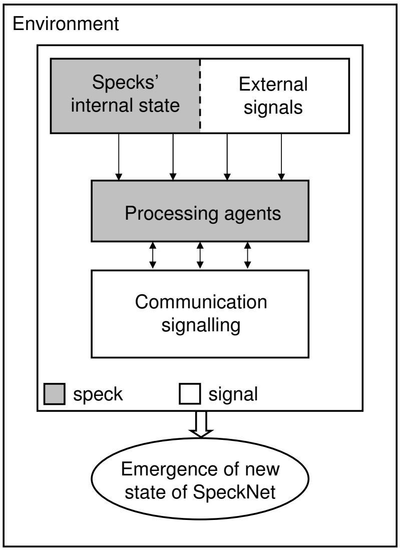


Fig. 4. Initial generic framework for SpeckNet

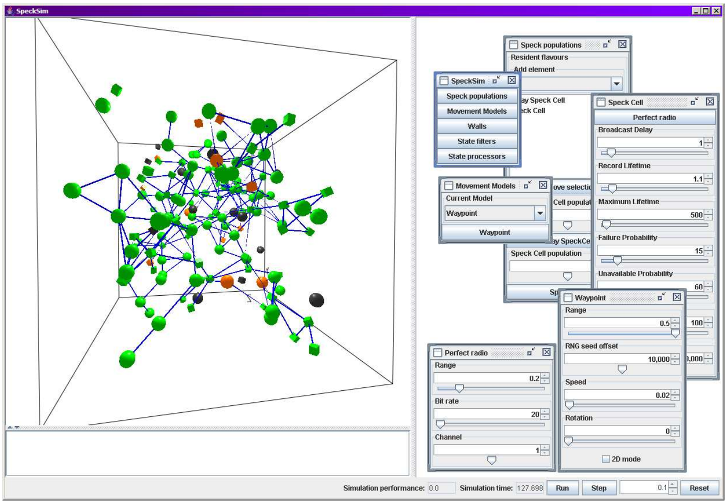
in dynamic, changing environments which provide them with additional input data, corresponding to antigen data in an immune system. For example, sensors on a speck may provide information regarding humidity, light, pressure, or temperature. This can be viewed as *external* information. External signals may ultimately affect the internal state of a speck, for example the environment may renew power supplies of specks (e.g. provide energy captured by solar cells) or may cause permanent damage to the network, by breaking some nodes.

Internal and external information is continuously gathered by each type of speck in a network. This information is filtered, integrated and processed by individual specks according to their designated *type* — similarly, internal and external signals are gathered and processed by a number of different types of cells in the immune system, e.g. macrophages, T cells and B cells. This requires the reception and generation of *signals* which are communicated to local specks and eventually may spread globally. As in the immune system, data is not generated, sent or received in a sequential or orderly fashion. Communication is asynchronous, thus the system is independent of signalling time scale. Collectively, specks undertake the responsibility of collating data (collect broadcasts), filtering signals (accept some, reject others), and processing them (transform them appropriately, e.g. calculate average value over a specific time window).

The framework allows the system to be continuously active and results in the emergence of a new state of the SpeckNet. Depending on the current conditions, this new state can be either of maintenance (e.g. specks work in a power saving mode), in which the aim of the system is to stay alive for the longest possible time, or of response, in which a noteworthy change in the environment activates the system and if necessary causes a reaction. Ongoing research is now directed at designing and testing types of specks to fulfil the roles outlined in this framework. Whilst overall functionality is achieved via vast numbers of unique cells in the natural immune system, this is clearly impractical in a SpeckNet work. Therefore, we intend to design speck types corresponding to the major classes (macrophages, T cells, B cells) which will be capable of processing internal and external information, and sending to, receiving and filtering messages from, other specks. In this first instance, this development work will be performed in simulation. Therefore, in the next section we outline a simulation tool which can be used to model and visualise the behaviour of large SpeckNets.

4.1 A SpeckNet Simulator: SpeckSim

Much of the development work of the proposed framework will take place using as a foundation the SpeckSim simulator developed by the SpeckNet research team [23]. This freely available Java-based simulator (see Fig. 5), incorporates several features, such as three-dimensional simulation and visualisation, asynchronous execution and graph file generation among others. Movement models incorporated in the design allow static or mobile arrangement of SpeckNets on a 2D plane or within 3D space. We are currently extending this simulator in order to facilitate the use of different types of specks and to provide a more realistic environment, in which a variety of signals can be generated to reflect the external environment, and in which individual specks are subject to the same unreliable events as real specks, e.g. randomly "dying" or becoming unavailable for extended periods of time. Investigation will then begin into methods of distributing functionality across different types of specks. For example, some specks



Speck Cell population

Current Model: Waypoint

Broadcast Delay: 1

Record Lifetime: 1.1

Maximum Lifetime: 500

Failure Probability: 15

Unavailable Probability: 60

100

0,000

Range: 0.5

RNG seed offset: 10,000

Speed: 0.02

Rotation: 0

□ 2D mode

Range: 0.2

Bit rate: 20

Channel: 1

Simulation performance: 0.0 Simulation time: 127.698 0.1

Fig. 5. SpeckSim, a tool for performing algorithm-level simulations on SpeckNets

may simply relay information (equivalent to cytokines) whereas others may filter and aggregate information. Simulations corresponding to a number of different case studies will then allow a range of scientific questions to be studied, regarding the performance and the quality of a SpeckNet, when operating under the immune-based framework. For example, we wish to establish to what extent the networked system is robust to failure of individual speck nodes; how sensitive the SpeckNet appears to be to its parameters, how much tuning it needs in order to work, and what are the boundaries within which it exhibits sensible behaviours.

5 Conclusion

In this paper we have presented an argument to show that the natural immune system, and in particular Cohen's perspective of a cognitive immune system, offers an appealing metaphor for the design of a framework to enable tiny and massively networked devices with very specific engineering constraints to self-organise and function in a reliable manner. A software platform is now being developed as part of the Speckled Computing project, and will be tested in simulation over the coming months.

The development work is still in its infancy, and clearly a number of questions arise that remain to be addressed. For example, the heterogeneity of the speck platform required, in terms of the distribution of speck types required for a functioning network to emerge is a key area to be investigated. Another area which raises a number of issues is that of communication between specks. Interactions in the immune network are without doubt the crucial operation in the ultimate emergence of a functioning network which maintains and protects the body, yet in real-world WSNs, communication is expensive and limited in terms of what can realistically be broadcast. Much of the work in the current project will be performed in simulation, yet ultimately the real test will lie in the transfer of the mechanisms to real networks, sometime in the future.

Nevertheless, the field of WSN (and the wider field of pervasive computing in general) provides an exciting and challenging research agenda for the future, for Speckled Computing and for the field of ubiquitous computing in general. Perhaps this domain may finally provide the field of AIS with the "killer application" that has remained so elusive over the last decade.

Acknowledgements

This research is funded by the SpeckNet consortium, funded by SHEFC HR03007.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Computer Networks (Elsevier) Journal 38(4), 393–422 (2002)
2. Andrews, P.S., Timmis, J.: Inspiration for the next generation of artificial immune systems. In: Proceedings of the 4rth International Conference on Artificial Immune Systems. Banff, Alberta, Canada, pp. 126–138 (2005)

3. MacGregor, F.: A spray-on computer is way to do IT. Article in The Evening News, The Edinburgh Paper August 14, (2003) (retrieved May 28, 2007), http://edinburghnews.scotsman.com/index.cfm?id=891382003
4. Cohen, I.R.: The cognitive principle challenges clonal selection. Immunology Today 13(11), 441–444 (1992)
5. Cohen, I.R.: The cognitive paradigm and the immunological homunculus. Immunology Today 13(12), 490–494 (1992)
6. Cohen, I.R.: Tending Adam's garden: evolving the cognitive immune self. Elsevier Academic Press, Amsterdam (2000)
7. Crockett, L.H., MacEwen, N.C., Pfann, E., Stewart, R.W.: A low power, digital transceiver for wireless sensor networks. In: Proceeding of the 2nd IEE/Eurasip Conference on DSP Enabled Radio, London, UK, pp. 18/1–18/6 (2005)
8. Dasgupta, D.: Advances in artificial immune systems. IEEE Computational Intelligence Magazine 1(4), 40–49 (2006)
9. de Castro, L.N., Timmis, J.: Artificial immune systems: a new computational intelligence approach, pp. 54–56. Springer, Heidelberg (2002)
10. Dressler, F., Krüger, B., Fuchs, G., German, R.: Self-organization in sensor networks using bio-inspired mechanisms. In: Proceedings of 18th ACM/GI/ITG International Conference on Architecture of Computing Systems - System Aspects in Organic and Pervasive Computing: Workshop Self-Organization and Emergence, Innsbruck, Austria, pp. 139–144. ACM Press, New York (2005)
11. Dressler, F.: Benefits of bio-inspired technologies for networked embedded systems: an overview. Dagstuhl Seminar 06031 on Organic Computing - Controlled Emergence (2006)
12. Estrin, D., Girod, L., Pottie, G., Srivastava, M.: Instrumenting the world with wireless sensor networks. In: International Conference on Acoustics, Speech, and Signal Processing, vol. 4, pp. 2033–2036 (2001)
13. Garrett, S.M.: How do we evaluate artificial immune systems? Evolutionary Computation 13(2), 145–177 (2005)
14. Hart, E., Timmis, J.: Application areas of AIS: the past, the present and the future. In: Proceedings of the 4th International Conference on Artificial Immune Systems, Banff, Alberta, Canada, pp. 483–497 (2005)
15. Heidemann, J., Govindan, R.: An overview of embedded sensor networks. Technical Report ISI-TR-2004-594, USC/Information Sciences Institute (2004)
16. IEEE Standard for Information Technology: IEEE Std 802.15.4-2003 (retrieved May 28, 2007), http://standards.ieee.org/getieee802/802.15.html
17. Kim, S., Pakzad, S., Culler, D., Demmel, J., Fenves, G., Glaser, S., Turon, M.: Poster abstract: wireless sensor networks for structural health monitoring. In: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, pp. 427–428 (2006)
18. Levis, P., Culler, D.: Maté: a tiny virtual machine for sensor networks. In: Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 85–95 (2002)
19. Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D., Anderson, J.: Wireless sensor networks for habitat monitoring. In: Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, pp. 88–97. ACM Press, New York (2002)
20. Martinez, K., Ong, R., Hart, J.: Glacsweb: a sensor network for hostile environments. In: First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, pp. 81–87. IEEE Computer Society Press, Los Alamitos (2004)

21. McNally, R., Wong, K.J., Arvind, D.K.: A distributed algorithm for logical location estimation in Speckled Computing. In: Proceedings of the IEEE Wireless Communications & Networking Conference, USA, vol. 3, pp. 1854–1859. IEEE Computer Society Press, Los Alamitos (2005)
22. Simon, G., Maróti, M., Lédeczi, Á., Balogh, G., Kusy, B., Nádas, A., Pap, G., Sallai, J., Frampton, K.: Sensor network-based countersniper system. In: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, pp. 1–12 (2004)
23. Speckled Computing Consortium (retrieved May 28, 2007), http://www.specknet.org
24. Wong, K.J., Arvind, D.K.: Speckled Computing: disruptive technology for networked information appliances. In: Proceedings of the IEEE International Symposium on Consumer Electronics, pp. 219–223. IEEE Computer Society Press, Los Alamitos (2004)
25. Wong, K.J., Arvind, D.K., Sharwood-Smith, N., Smith, A.: Specknet-based Responsive Environments. In: Proceedings of the IEEE International Symposium on Consumer Electronics, Macau, pp. 334–338. IEEE Computer Society Press, Los Alamitos (2005)
26. Whyte, G., Buchanan, N., Thayne, I.: An omnidirectional, low cost, low profile, 2.45 GHz microstrip fed rectaxial antenna for wireless sensor network applications. In: IEE and IEEE Loughborough Antennas and Propagation Conference, IEEE Computer Society Press, Los Alamitos (2006)
27. Young, A., Ling, M., Arvind, D.K.: Orient-2: A realtime wireless posture tracking system using local orientation estimation. In: Proceedings of the Fourth Workshop on Embedded Networked Sensors, 2007 (to appear)
28. Zambonelli, F., Van Dyke Parunak, H.: Signs of a revolution in computer science and software engineering. In: Engineering Societies in the Agents World III: Third International Workshop, Madrid, Spain, pp. 13–28 (2002)

Biological Inspiration for Artificial Immune Systems

Jamie Twycross and Uwe Aickelin

School of Computer Science, University of Nottingham, UK
jpt@cs.nott.ac.uk

Abstract. Artificial immune systems (AISs) to date have generally been inspired by naive biological metaphors. This has limited the effectiveness of these systems. In this position paper two ways in which AISs could be made more biologically realistic are discussed. We propose that AISs should draw their inspiration from organisms which possess only innate immune systems, and that AISs should employ systemic models of the immune system to structure their overall design. An outline of plant and invertebrate immune systems is presented, and a number of contemporary systemic models are reviewed. The implications for interdisciplinary research that more biologically-realistic AISs could have is also discussed.

1 Introduction

The field of Artificial Immune Systems began in the early 1990s with a number of independent groups conducting research which used the biological immune system as inspiration for solutions to problems in non-biological domains. Since that time artificial immune system (AIS) research has produced a considerable body of knowledge and a number of general purpose algorithms. AISs based on these algorithms have been applied to many benchmark and a number of real-world problems. Currently however, the field is at an impasse [1,2]. While there have been some success stories on realworld problems, there is still little to differentiate the performance of AISs with other state-of-the-art methods. We concur with Timmis [2] that this is due to a limited application to challenging problems, a lack of theoretical advances, and the use of naive biological metaphors. In this position paper we focus on biological metaphors and discuss the areas of biology that we believe should be important in inspiring future AISs. Our intention is to draw the attention of AIS researchers to these areas and to provide references to key papers which we have found useful in understanding the biology.

This paper argues that AISs can be made more biologically realistic in two ways. In the first place, we believe that AIS researchers should consider drawing inspiration from simpler biological systems than humans. A serious evaluation of the validity and usefulness of building AISs inspired by the adaptive immune system needs to take place. The vast majority of life survives and flourishes without an adaptive immune system. The innate immune system mechanisms employed by the majority of organisms provide robust maintenance of organism integrity and protection against pathogens. While complex, these purely

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 300–311, 2007.

innate immune systems are relatively simpler in organisational terms than immune systems which combine both innate and adaptive arms. Recent research has also shown that innate immune systems exhibit properties such as specificity, diversity and memory, previously only associated with adaptive immune systems. Innate immune systems can and do do everything adaptive immune systems do, including adapt to rapidly evolving pathogens, albeit using different mechanisms [3]. It seems only sensible to start with simpler innate-based AISs before building adaptive immune mechanisms into AISs.

Secondly, AISs need to be based around more contemporary and sophisticated systemic models of the immune system than those currently employed. As shown by these contemporary models, the view of the immune system as a protective system driven by adaptive immune system mechanisms of self/nonself discrimination is at odds with current immunological thinking on how the immune system behaves as a complete system. While self/nonself discrimination is a characteristic observed in both innate and adaptive immune systems, it is not the purpose of the immune system. Yet, as a survey of past ICARIS proceedings [4] reveals, the majority of AISs built so far have been built for the purpose of discriminating self from nonself. This is not just arguing over semantics, but goes to the heart of the engineering philosophy used to build AISs.

Even if we must build AISs which incorporate adaptive immune system mechanisms, it makes little sense to build them based only the adaptive immune system. There is no organism in existence with only an adaptive immune system. Organisms which do possess an adaptive immune system also have innate immune systems. There seems to us to be a very good reason for this. While the adaptive immune system provides the organism with a diverse set of receptors which can recognise almost any molecule, it provides very little control over this recognition. The control of the adaptive immune system is firmly in the hands of the innate immune system [5]. Building AISs which model only adaptive immune system mechanisms is like building a car without a steering wheel - it will certainly go somewhere, but you have very little control as to where this is!

In the first part of this paper we discuss current understanding of the immune systems of plants and invertebrates with the idea that these organisms could provide simpler biological systems from which to draw inspiration for AISs. In the second part of this paper we discuss systemic models of the human immune system. In particular, in light of the first part of this paper and the importance of the innate immune system, we focus on systemic models which are concerned with how the innate and adaptive immune systems are integrated. The paper ends with a brief discussion of the implications for interdisciplinary research that more biologically-realistic AISs could have.

2 Non-human Immune Systems

The majority of AISs to date have been inspired by vertebrate adaptive immune system mechanisms. This focus of AIS research on the adaptive immune system is in some ways similar to Artificial Intelligence's early focus on the human mind and symbolic information processing. Only more recently has the focus of AI been expanded by the acknowledgement of intelligence in the wider sense of adaptive behaviour of organisms other than humans. We firmly believe that the field of AISs also needs to reassess its sources of biological inspiration and focus on the immune systems possessed by the majority of life on this planet. The adaptive immune system may be interesting and useful, but is in no way a prerequisite for a successful immune system, just as playing chess is interesting and useful but is in no way a hallmark of intelligent behaviour. By studying plant and invertebrate immune systems, differences and commonalities that exist between immune systems can also be uncovered. This could well help identify general principles of immune systems which could be of use to AIS researchers.

2.1 Plant Immune Systems

Plants do not have specialised defender cells and rely on innate immunity provided by each cell in the plant. Upon infection with a pathogen, plant cells are induced to produce a range of antimicrobial products which help neutralise pathogens. Pathogens which survive usually trigger a hypersensitive cell death response (HR), which causes host cells at the site of infection to die. Both HR and production of antimicrobial products need to be tightly controlled and plants have evolved intricate systems to do this. Inducible plant immunity is provided by two different but interacting systems. The first system is based around pattern recognition receptors (PRRs) on the surface of plant cells. These PRRs are activated by molecules produced by pathogens called pathogen- or microbial-associate molecular proteins (PAMPs or MAMPs). The second system, which acts intracellularly, is based around a set of polymorphic proteins called nucleotide-binding site plus leucine-rich repeat (NB-LRR) proteins. These NB-LRR proteins are coded for in the genome of the plant by specific disease resistance (R) genes [6].

Inducible immunity in plants is currently viewed as a four-phase process. Phase 1 is initiated by the recognition of PAMPs or MAMPs by PRRs and induces a set of responses known as PAMP-triggered immunity (PTI). In Phase 2, pathogens which succeed in overcoming the initial PTI response produce effector molecules (also known as virulence factors) which enhance the spread of the pathogen and can also suppress PTI responses. Phase 2 results in effector-triggered susceptibility of the host to the pathogen. In Phase 3, effectors produced by the pathogen are recognised by NB-LRR proteins encoded by R genes and initiate effector-triggered immunity (ETI). The specific effector which is recognised is termed an avirulence (Avr) protein. ETI responses are amplified versions of PTI responses and usually result in the death of the infected host cell. In Phase 4 both host and pathogen undergo a process of selection in which pathogen variants which do not produce the triggering Avr protein but instead produce other effectors are selected for. At the same time host R genes which produce NB-LRR proteins which recognise the new effectors are selected for, once again resulting in ETI [6,7].

While direct recognition of Avr proteins by NB-LRR proteins has been observed, indirect recognition of Avr proteins also occurs. In indirect recognition, NB-LRR proteins are activated by products of the action of Avr proteins on the host. The 'guard hypothesis' has been proposed as a conceptual framework to explain indirect recognition. Pathogen Avr proteins target specific host molecules in order to increase the spread of the pathogen. Host NB-LRR proteins guard these molecules and are activated by changes in their guardees caused by pathogens. NB-LRR proteins either constitutively bind to their guardees and disengage and are activated when pathogen Avr proteins interact with the guardee. Alternatively, NB-LRR proteins are activated by the molecular complex produced when the Avr protein binds with the guardee [6,7].

As well as the protective mechanisms targeted at pathogens such as bacteria, viruses and fungi just described, plants also possess an array of mechanisms designed to protect them against herbivores such as insects and mammals. These mechanisms are triggered by wounding of the plant by herbivores which causes the production of both direct and indirect defences which are often tailored to the attacking herbivore. Direct defences include the release of antidigestive proteins which reduce the performance of the herbivore by interfering with its digestive enzymes, and the release of antinutritive enzymes which decrease the nutritional value of the plant. Indirect defence mechanisms result in the production of volatile organic compounds (VOCs). These VOCs attract herbivore predators and parasites, and allow top-down control of herbivore populations [8].

Lastly, plants possess systems that are unique among recognition systems in that they produce responses that are the converse of immune responses. Recognition of self (the same plant) produces a response, and nonself (a different plant) does not produce a response [9]. Hermaphroditic plants which produce both pollen and pistel have developed recognition systems to prevent inbreeding, that is, fertilisation of the plant by itself. These self-incompatibility (SI) systems allow plant species to maintain genetic diversity. SI systems depend upon a set of highly polymorphic genes called the S locus which code for both an S-locus receptor protein kinase (SRK) and an S-locus cysteine-rich (SCR) ligand. The SRK receptor is present on the pistel, while the SCR ligand appears on pollen. Binding of SCR to SRK from the same S locus i.e. the same plant, activates the SRK receptor and leads to the arrest of fertilisation, whereas SCR derived from S loci of different plants does not activate the SRK receptor and allows pollination to proceed [10].

2.2 Vertebrate and Invertebrate Immune Systems

Around 97% of all animal species are invertebrates and have no adaptive immune system. Yet their immune systems have evolved to help make them the most prolific animals on the planet. Invertebrate immune systems are "*not homogeneous, not simple, not well understood*" [11]. Studies of invertebrate immune systems have demonstrated that, while only possessing innate immune systems, their immune systems still exhibit phenomena such as specificity, diversity and memory which were previously only associated with the adaptive immune system. For example, in mosquitoes, Down syndrome cell adhesion molecule (Dscam) has been identified as having characteristics similar to human immunoglobulin and is able to produce a diverse set of over 30,000 proteins which enable specific recognition of bacteria. Diversity of Dscam proteins is produced in a similar way to vertebrate immunoglobulin through somatic rearrangement of Dscam gene segments [12,13]. Invertebrates have also been shown to exhibit specific memory, that is, enhanced protection against the same pathogen upon reinfection. Long-lasting upregulation of regulatory pathways and production of stable proteins such as fibrogen-related proteins (FREPs) in snails have been proposed as mechanisms of specific memory in invertebrates [14,15].

Evolution has taken different routes to achieve functionally similar systems. In other words, both invertebrate and vertebrate immune systems have evolved different mechanisms which provide antigen-specific memory and protection. Immunoglobulin-based adaptive immune systems have been identified in almost all jawed vertebrates, but not in jawless vertebrates or invertebrates [16]. Lampreys and hagfish, both jawless vertebrates, do not produce immunoglobulin, but instead generate their own diverse set of proteins called variable lymphocyte receptors (VLRs) in response to invading microbes. Lampreys can generate up to 100 trillion unique VLRs. VLRs are made up of proteins called leucine-rich repeat (LRR) modules, and their diversity is generated by a process of somatic rearrangement of LRR modules which surround a single VLR gene [17,18]. This process of protein rearrangement contrasts with the generation of T cell receptors by somatic recombination of multiple VDJ gene segments in jawed vertebrates. For a review of immune system mechanisms from an evolutionary perspective in invertebrates, protochordates, and jawed and jawless vertebrates see [3].

Thus, while both jawed and jawless vertebrates possess an adaptive immune system, the underlying components and processes of their systems have evolved in different ways. And while invertebrates have no adaptive immune system, they have evolved innate immune systems which provide similar functionality to vertebrate adaptive immune systems. The end result is the same - the production of a diverse set of proteins that provide the host with a mechanism of specific recognition, diversity and memory. The commonalities between Dscam, VLR and immunoglobulin molecules could provide important insights into the essential properties which AISs need to reproduce in their artificial T cell receptors. The differing somatic and germline rearrangement mechanisms which are involved in the generation of Dscam, VLR and immunoglobulin diversity could, for example, provide inspiration for new AIS gene library algorithms.

3 Systemic Models of the Human Immune System

Systemic immunological models explore how systemic properties such as immunity and tolerance are generated by the immune system as a whole. The immune system is at this level a system, an assemblage of different interacting entities which comprise a whole. Essentially, systemic models seek to answer questions about what the immune system does and how it does it. Obviously, an understanding of such models is essential for computer scientists seeking to build AISs which exhibit similar systemic properties to the biological immune system. However, the majority of AISs to date have been based on the assumption that the overall purpose of the immune system is to protect the host, and that it does so by mechanisms based around self/ nonself discrimination. Adoption of more sophisticated and realistic contemporary models is necessary if AISs are to prove successful at solving hard realworld problems. These models are discussed further in relation to AISs in [19,20].

Over the course of several decades immunologists have developed a number of systemic models of immunity. For a historical overview and comparison of some of these models see [21,22]. Many of the more contemporary models are discussed by their protagonists in the internet-based "*The Great Debate: The web debate on self-nonself*" [23], in which, over a period of five days, leading immunologists debate these models via email and offer some keen insights into their similarities and differences. These models have reflected and guided experimental research. Sakaguchi [24] characterises immunological research in terms of two ancient Greek mottos of Delphi: "*Gnothi Seauton*" (know thyself) and "*Meden Agan*" (nothing in excess). He contends that while 'know thyself' has been a favourite slogan of immunologist for many years, the important of 'nothing in excess' has received relatively little attention. The latter truth, manifested in immune homeostasis and self-tolerance, is however, a vital principle of immunity. In this section, we briefly overview these various systemic models and present a categorisation in terms of the way these models view the relationship of the immune system to the body and to itself. In essence, models can be categorised as to whether they see the immune system as a protector or maintainer of the body or of itself. One common feature of contemporary models is the central role they give to the innate immune system as controller of the immune system.

3.1 Self/Nonself Discrimination

It has been long been observed that when pathogens, destructive microorganisms such as viruses or bacteria, enter the body, the immune system removes them and returns the body to a healthy state. Naturally then, the purpose of the immune system is often seen as that of a protector or defender of the body. Since the immune system reacts to pathogens (nonself in immunological terms) but not to the body (self), it also seems logical to conclude that the immune system provides this protection by discriminating self from nonself. Defence by self/nonself discrimination has formed the basis of the majority of immunological models since the middle of the last century, and this view of the immune system is still widely accepted by immunologists today [25].

Earlier models of immunity were based around the idea that host constituents (self) are ignored by the immune system, while other elements (nonself), such as pathogens, foreign substances or altered self, are reacted to. In these models, tolerance is largely viewed as immune system silence or nonreactivity to self. Models such as Burnet's Clonal Selection Theory [26] and the Associative

Recognition Model of Bretscher and Cohn [27] rest on a historical mechanism in which immature receptor-bearing cells of the adaptive immune system are exposed to a wide range of non-pathogenic material early in the development of the organism. If this non-pathogenic material is recognised above a certain level, this leads to the destruction of the cell and its receptors. This results in a set of mature cells whose receptors only recognise antigen which are not historically part of the organism. This recognition leads to the initiation of an immune response and the destruction of the pathogen to which the antigen belongs.

3.2 Infection and Danger

Other models, based on divisions of antigen other than self/nonself, have also been developed. For these models, the immune system does not partition the antigenic universe into two groups of self and nonself molecules. Self/nonself discrimination has been criticised for being applied to the mechanisms which produce overall immune system behaviour. It has been observed [28] that referring to self/nonself discrimination of antigen by T cells is a category error. While the immune system as a whole appears to recognise self from nonself (a systemic property) this does not imply that individual T cells recognise self antigen. This is making the mistake of attributing the property of a system to its elements. While preserving the protective purpose of the immune system, models such as the Infectious Nonself Model and Danger Model have moved away from self/nonself discrimination as the driving force behind immunity.

The Infectious Nonself Model of Janeway [29] like earlier models, views the purpose of the immune system as protecting the body. However, the Infectious Nonself Model proposes that instead of categorising antigen into self and nonself, the immune system categorises antigen into the classes of infectious nonself and noninfectious self. Moreover, instead of the adaptive immune system based historical process of negative selection, detection of pathogens by innate immune system cells is seen as the principal controller of the immune system. Janeway proposes that innate immune system antigen presenting cells (APCs), especially dendritic cells, are the principal controllers of the immune system. In a similar way to plant cells as discussed in Section 2.1, APCs express a groups of receptors called pattern recognition receptors (PRRs) which respond to pathogen-associated molecular patterns (PAMPs). Janeway defines PAMPs as "*conserved molecular patterns that are essential products of microbial physiology ... unique to microbes ... [and] not produced by the host ... [which] are recognized by receptors of the innate immune system ... [and which] induce the expression of costimulatory molecules on the cell [APC] surface, which is necessary for the activation of naive T cells*" [30]. Activation of PRRs by PAMPs initiates and modulates an immune response, and the activation of different subsets of PRRs tailors the immune response to different classes of pathogens.

The Danger Model of Matzinger [31,32] is similar to the Infectious Nonself Model, viewing the immune system as a protector, and the innate immune system as having a central role in the generation of protection. It also agrees that APCs have PRR receptors which when bound to certain molecules, activate the

APC, allowing it to express antigen in a stimulatory fashion. However, instead of being specific for material associated with pathogens, these receptors are specific for molecules, termed danger signals, produced when the tissue of the organism is damaged or stressed. Matzinger defines danger signals as "*a set of molecules elaborated or released by stressed or damaged cells, for which resting APCs have receptors, and to which resting APCs respond by becoming activated and upregulating costimulatory capacity*" [33]. Danger signals are released by cells when they undergo necrosis, unprogrammed death, but not when they undergo apoptosis, cell death which occurs as part of the normal functioning of the organism.

Although both models appear similar, their explanations of the origin of the material which activates APCs, and hence is responsible for the activation of an immune response, lead to important differences. By proposing host damage as the main regulator of the immune system the Danger Model expands the definition of the innate immune system to include the tissue cells of the host itself. In fact, these tissue cells are the cells that control innate immunity, and the class of immune response is not determined by the pathogen but rather by these tissue cells themselves. A key similarity between these models is the shift in control of the immune system from the T and B cells of the adaptive immune system to the cells of the innate immune system.

3.3 Maintenance and Homeostasis

Notions of dangerous and harmless are, however, themselves problematic. Philosophers such as Canguilhem [34] and Haraway [35] have observed how the concepts of pathological and normal are just as metaphorical and observer-dependant as those of self and nonself. Is the pathological an overexpression of the normal (a hyperreaction) or is it a radically different state from the normal? What exactly does the normal or average state mean? How have wider social and scientific notions of self and nonself influenced the way the immune system is understood? Other models have emerged which challenge the view of the immune system as a purely defensive and discriminatory system, and widen its functions to include host-maintenance and self-assertion or homeostasis.

Models such as these generally reject the notion that recognition equals pathogenicity. Instead, there is constant recognition and reaction by the immune system, which leads to a tolerogenic or immunogenic response. In these models, in place of a defender, the immune system is viewed as a maintenance or homeostatic system, maintaining the body or itself respectively. Resistance to change, for example produced by a pathogen, results in behaviour that appears to protect the body and recognise the pathogen. But it is the maintenance of the body in a particular state that is really the driving force behind this behaviour. Some models go further and assert not only that the purpose of the immune system is maintenance, but that it is self-maintenance or self-assertion, and not body maintenance. These models view the immune system as a homeostatic system, an open system which regulates its internal environment and maintains a state of dynamic equilibrium in the face of changes in its environment.

Cohen's cognitive paradigm [36,37] describes the immune system as a cognitive systems in which a dialogue is constantly taking place between immune cells and the body. Interactions between, for example, APCs and T cells can be described in terms of APCs communicating sentences describing the nature of an antigen to T cells. The subject of the sentence is the antigen. The predicate is a complex set of costimulatory molecules and cytokines produced by the surrounding tissue, or by APCs in response to signalling through germline innate receptors. The immune meaning of an antigen is defined as how the T cells responds to this sentence, with the context of the antigenic subject provided by the predicate. Through continuous dialogue between immune cells and the host, the immune system generates an internal image of self, which Cohen terms the immunological homunculus. Andrews and Timmis discuss the cognitive paradigm in relation to AISs further in [38].

The Morphostasis Model of Cunliffe [39,40] is based on the idea that the function of the immune system is tissue homeostasis. All cells in the body are able to sense when their normal function is disrupted, for example through co-option by a virus. When this occurs, cells signal this abnormality to neighbouring cells and sometimes apoptose. However, many pathogens have developed the ability to prevent their target cells from apoptosing. Phagocytic innate immune system cells such as macrophages and neutrophils play a key role in the Morphostasis Model. Phagocytes are able to sense changes in the normal functioning of cells and remove these cells. In this model the role of the adaptive immune system is to accelerate the identification and clearance of non-healthy cells by phagocytes. The Integrity Model of Dembic [41,42] is similar to the Morphostasis Model in that it characterises the immune system as maintaining the body through surveillance of the state of tissue. In the Integrity Model innate immune system dendritic cells scan tissue and detect changes in signal levels produced by tissue cells. This induces dendritic cells to initiate an adaptive immune system response.

4 Conclusion

In this paper we have argued for the need for AISs which are based on much more biologically-realistic models. We argued that, instead of building AISs based on the extremely complex human adaptive immune system, AISs should draw inspiration from relatively simpler organisms which possess only innate immune systems. In the first half of this paper we outlined current biological understanding of plant and invertebrate immune systems. The innate immune systems of these organisms are capable of self/nonself discrimination, and also exhibit properties such as specificity, diversity and memory which until recently have only been associated with adaptive immune systems. Low-level biological models of the mechanisms which give rise to these properties could provide important sources of inspiration for future AIS algorithms. If AISs are however to employ adaptive immune system mechanisms, then we argued that they also need to incorporate innate immune system mechanisms, which control the adaptive immune system in biological organisms. In the second half of this paper we outlined a number of systemic models of the human immune system which deal with how the innate and adaptive immune systems are integrated. These models provide AIS designers with a concrete framework for incorporating innate and adaptive immune mechanisms into their artificial systems.

As well as producing more effective AISs, building AISs based on more biologically-realistic models could have important consequences for biological research. For a number of years we have had the opportunity to work closely with immunologists. During this time we have been keen to develop interdisciplinary relationships which have benefited these immunologist as much as they have benefited us. Realistically however, this has proved very difficult, and we feel that we, and the field of AISs in general, have had very little impact on immunological research and thinking. Part of the reason for this is that the naive models employed by AISs have borne little resemblance to the models of immune system mechanisms employed by immunologists. Perhaps a more fundamental reason for this state of affairs is that the focus of immunological and AIS research often differ. Immunology has been largely focussed on elucidating the cellular and molecular basis of the immune system using a reductionist methodology. The field of AISs on the other hand is often concerned with building complete systems and adopts a more holistic methodology. An exception to this within Immunology is Systems Immunology, which studies how entities and mechanisms interact at different system levels to determine immune system behaviour, and whose domain includes systemic models of the immune system. Here we believe that AISs, by building artificial systems based on more biologically-realistic systemic models of the immune system, could have a significant impact. Such AISs, when applied to complex realworld problems, could provide important experimental systems which could be more easily manipulated and from which data could be more easily gathered than biological systems. These AISs could then be used to validate systemic immune system models. In this way, AIS research could have a real impact on Immunology.

AIS research to date has largely been concerned with engineering, that is, building useful machines or systems which solve practical problems. Whether or not the immune-inspired principles used reflect any fundamental properties of biological immune systems is of little consequence. If they are useful in achieving the practical ends of the engineer then they have served their purpose well. This engineering approach, in our opinion, while being productive in developing solutions to practical problems, has further limited the interdisciplinary impact of AIS research. What is needed to address this limitation is an expansion of the scope of AIS research to address fundamental questions in Immunology and the organisation of complex systems. This can only be done if, on the one hand, more biologically-realistic models are adopted and studied specifically to understand the dynamics of these models, and whether they capture the dynamics of the biological systems they seek to describe. On the other hand, insights from computer science into complex systems and techniques for modelling such systems could help immunologists to develop better biological models. Perhaps what we will see in the future is AISs grow into a field which takes its biology as seriously as its engineering. In this case, a more appropriate definition of the field would be Artificial Immunology - the construction and study of immune-systems-as-they-could-be in an effort to understand immune-systems-as-they-are and to enhance the construction of immune systems for artificial organisms.

Acknowledgements

This research is supported by EPSRC Grant GR/S47809/01.

References

1. 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, pp. 483–497. Springer, Heidelberg (2005)
2. Timmis, J.: Artificial immune systems - today and tomorrow. Natural Computing 6(1), 1–18 (2007)
3. Litman, G.W., Cannon, J.P., Dishaw, L.J.: Reconstructing immune phylogeny: new perspectives. Nature Reviews in Immunology 5, 866–879 (2005)
4. In: Proceedings of the International Conference on Artificial Immune Systems. LNCS, Springer, Heidelberg (2002-2006), `http://www.artificial-immune-systems.org/`
5. Germain, R.N.: An innately interesting decade of research in immunology. Nature Medicine 10(12), 1307–1320 (2004)
6. Jones, J.D.G., Dangl, J.L.: The Plant Immune System. Nature 444, 323–329 (2006)
7. Dangl, J.L., Jones, J.D.G.: Plant pathogens and integrated defence responses to infection. Nature 411, 826–833 (2001)
8. Kessler, A., Baldwin, I.T.: Plant responses to insect herbivory: the emerging molecular analysis. Annual Review of Plant Biology 53, 299–328 (2002)
9. Boehm, T.: Quality Control in Self/Nonself Discrimination. Cell 125(5), 845–858 (2006)
10. Nasrallah, J.B.: Recognition and Rejection of Self in Plant Reproduction. Science 296, 305–308 (2002)
11. Loker, E.S., Adema, C.M., Zhang, S.M., Kepler, T.B.: Invertebrate immune systems - not homogeneous, not simple, not well understood. Immunological Reviews 198, 10–24 (2004)
12. Kurtz, J., Armitage, S.A.: Alternative adaptive immunity in invertebrates. Trends in Immunology 27(11), 493–496 (2006)
13. Dong, Y., Taylor, H.E., Dimopoulos, G.: AgDscam, a Hypervariable Immunoglobulin Domain-Containing Receptor of the Anopheles gambiae Innate Immune System. PLoS Biology 4(7), 1137–1146 (2006)
14. Zhang, S.-M., Adema, C.M., Kepler, T.B., Loker, E.S.: Diversification of Ig Superfamily Genes in an Invertebrate. Science 305(5681), 251–254 (2004)
15. Kurtz, J.: Specific memory within innate immune systems. Trends in Immunology 26(4), 186–192 (2005)
16. Litman, G.F., Anderson, M.K., Rast, J.: Evolution of antigen binding receptors. Annual Review of Immunology 17, 109–147 (1999)
17. Pancer, Z., Saha, N.R., Kasamatsu, J., Suzuki, T., Amemiya, C.T., Kasahara, M., Cooper, M.D.: Variable lymphocyte receptors in hagfish. PNAS 102(26), 9224–9229 (2005)
18. Alder, M.N., Rogozin, I.B., Iyer, L.M., Glazko, G.V., Cooper, M.D., Pancer, Z.: Diversity and function of adaptive immune receptors in a jawless vertebrate. Science 310(5756), 1970–1973 (2005)

19. 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)
20. Twycross, J.: Integrated Innate and Adaptive Artificial Immune Systems applied to Process Anomaly Detection. PhD thesis, School of Computer Science, University of Nottingham, U.K (2007)
21. Langman, R.E., Cohn, M.: A short history of time and space in immune discrimination. Scandinavian Journal of Immunology 44(3), 544–548 (1996)
22. Matzinger, P.: Essay 1: the Danger model in its historical context. Scandinavian Journal of Immunology 54(1-2), 4–9 (2001)
23. HMS Beagle: The Great Debate: The web debate on self-nonself. Salk Institute (1997), http://cig.salk.edu/bicd_140_W99/debate/debate.htm
24. Sakaguchi, S.: Regulatory T cells: Meden Agan. Immunological Reviews 212(1), 5–7 (2006)
25. Pradeu, T., Carosella, E.D.: The Self Model and the Conception of Biological Identity in Immunology. Biology and Philosophy 21(2), 235–252 (2006)
26. Burnet, F.M.: The Clonal Selection Theory of Acquired Immunity. Vanderbilt University Press (1959)
27. Bretscher, P., Cohn, M.: A Theory of Self-Nonself Discrimination. Science 169, 1042–1049 (1970)
28. Tauber, A.I.: The elusive immune self: a case of category errors. Perspectives in biology and medicine 42(4), 459–474 (1999)
29. Janeway, C.A.: Approaching the asymptote? Evolution and revolution in immunology. Cold Spring Harbor Symposia on Quantitative Biology LIV, 1–13 (1989)
30. Medzhitov, R., Janeway, C.A.: Decoding the patterns of self and nonself by the innate immune system. Science 296, 298–300 (2002)
31. Matzinger, P.: Tolerance, Danger, and the Extended Family. Annual Review of Immunology 12, 991–1045 (1994)
32. Matzinger, P.: Friendly and dangerous signals: is the tissue in control?. Nature Immunology 8(1), 11–13 (2007)
33. Anderson, C., Matzinger, P.: Anderson and Matzinger: Round 3. Seminars in Immunology 12, 331–341 (2000)
34. Canguilhem, G.: The Normal and the Pathological. Zone Books (1991)
35. Haraway, D.: The Biopolitics of Postmodern Bodies: Constitutions of Self in Immune System Discourse. In: Simians, Cyborgs, and Women: The Reinvention of Nature, pp. 203–230. Free Association Books (1991)
36. Cohen, I.R.: The cognitive principle challenges clonal selection. Immunology Today 13(11), 441–444 (1992)
37. Cohen, I.R.: Tending Adam's Garden: Evolving the Cognitive Immune Self. Academic Press, London (1999)
38. 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)
39. Cunliffe, J.: Morphostasis and immunity. Medical Hypotheses 44(2), 89–96 (1995)
40. Cunliffe, J.: Tissue Homeostasis and Immunity - More on Models. Scandinavian Journal of Immunology 64(3), 172–176 (2006)
41. Dembic, Z.: Do we need integrity? Scandinavian Journal of Immunology 44(3), 549–550 (1996)
42. Dembic, Z.: Immune system protects integrity of tissues. Molecular immunology 37(10), 563–569 (2000)

Regulatory T Cells: Inspiration for Artificial Immune Systems

T.S. Guzella[1], 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 conceptual paper, some features of regulatory T cells are described. These cells have been receiving an increasing attention in Immunological research, due to their importance in several aspects of the immune system. As will be argued, these cells constitute an important source of inspiration for developing Artificial Immune Systems, computational tools that attempt to capture some of the characteristics of the natural immune system. It is expected that the incorporation of these cells in some immune inspired algorithms may not only lead to more biologically plausible models, but also to algorithms that can achieve better results in real-life problems.

Keywords: Artificial Immune System, Regulatory T cells, Conceptual models, Cross-regulation model.

1 Introduction

Recently, Artificial Immune Systems (AISs), have emerged as a novel soft computing paradigm [1], exploring some of mechanisms and components of the immune system. Due to the fact that AISs are a relatively new area, it is interesting to pursue a greater understanding of the biological models and mechanisms, which can serve as inspiration to the development of new algorithms or the improvement of some of the existing approaches.

There has been several papers dedicated to discussing the current state of AISs from a critical point of view. Garrett [2] has analyzed how distinct and effective immune based algorithms are, in comparison with other approaches, focusing on clonal selection, immune network, negative selection and danger-theory algorithms. Hart and Timmis [3] have discussed the need to reflect on the role that AISs can play, and also on the application areas where the potential of these algorithms can be asserted. Freitas and Timmis [4] have analyzed AISs from a problem-oriented perspective for data mining applications, focusing on the need to incorporate application-specific knowledge. In addition, they have commented on some important aspects of the natural immune system that are considered

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 312–323, 2007.

by few or no AISs, such as the two-signal model, and the immunoglobulin class switching. Dasgupta [5] has commented on the recent advances in AISs, focusing on some of the models that have inspired their development and applications tackled by current approaches.

In this work, the focus is on yet another aspect: the natural immune system. The main objective is to point out the importance of a special type of cell, referred to as regulatory (or suppressor) T cells, in several aspects of the natural immune system. As a matter of fact, understanding the role of these cells constitutes an area of intense research in Immunology. In contrast, these cells are rarely considered in the area of Artificial Immune Systems. By stating what is currently known about these cells from experimental systems, along with theoretical efforts to build mathematical models and hypothesis, this paper suggests the incorporation of some features of these cells in immune inspired algorithms.

This paper is organized in the following way: section 2 discusses in a moderate level of detail the roles of regulatory T cells in various aspects of the natural immune system, presenting some of the experimental systems that have lead to the affirmation of the importance of these cells. Following this, section 3 reviews some modelling studies that have focused on these cells, while section 4 comments on how some aspects of these cells are being incorporated into two models under development. Finally, section 5 presents the final conclusions of this paper and some of the future investigations to be conducted.

2 Regulatory T Cells (Tregs) in Immunological Research

An important aspect of the immune system is its ability to react against external, harmful agents (nonself or pathogens), while, most of the times, remaining unresponsive to self (self-tolerance). The importance of this discrimination by the immune system is twofold: first, because an immune response initiated against self components (giving rise to autoimmune diseases, or autoimmunity) could be devastating, and even fatal, for the host; on the other hand, if the system fails to initiate a response against a harmful pathogen, the host's integrity is under great risk.

The adaptive immune system, composed of B and T lymphocytes, is capable of identifying a large variety of antigens, molecules that can be recognized by specific lymphocyte receptors and antibodies. The two types of lymphocytes have distinctive roles: B cells are responsible for secreting antibodies, while helper T lymphocytes can activate senescent B cells, cytotoxic T cells eliminate host cells infected by intracellular pathogens, and regulatory T cells (Tregs) directly regulate the activation of other B and T cells. In contrast to B cells, which can bind directly to antigens, T cells require that peptides (small antigenic features) coupled to MHC (Major Histocompatibility Complex) are presented by antigen presenting cells (APCs), in the case of helper T cells, or by most cells of the body (for cytotoxic T cells) [6]. Because the receptors for B and T lymphocytes, which acquire immunocompetency in the bone marrow and thymus, respectively, are generated through a random gene rearrangement process, it is necessary to ensure that the produced lymphocytes will not initiate a response directed towards self antigens. The Clonal Selection Theory [7] has been, for almost 50 years, the dominating base to explain how the immune system is able to differentiate between self and nonself. After subsequent refinements over the years, the theory assumes that a process denominated negative selection, that takes place during the production of B and T cells, eliminates all such lymphocytes that sustain high affinity interactions with self antigens presented on the bone marrow and thymus, respectively. In addition, all self-reactive cells that would eventually escape from this deletion mechanism would be rendered unresponsive to stimuli (anergic), unable to respond to antigenic challenge. In addition, anergy could also be induced in the periphery (lymphoid and non-lymphoid tissues) under certain circumstances, as reviewed by Schwartz [8]. Further details on the historical perspective of these refinements can be found in [9].

However, there are several problems in some of the affirmations of the Clonal Selection Theory. As commented by Grossman and Paul [10], the concept of anergy as a mechanism to assert self-tolerance is problematic, as the number of such cells should be reduced to a minimum, due to the possibility of these cells being activated and causing autoimmune diseases. This is controversial with experimental evidence for the abundance of such cells. In addition, Coutinho [11] comments on two additional problems, referred to as the time and space problem. The space problem is due to the fact that, to ensure tolerance, it would necessary that every developing lymphocyte met every self antigen during negative selection, which is not possible, due to the large number of such antigens and the inability of keeping a copy of every one of them in the thymus or bone marrow. In addition, the time problem would occur because, as tolerance is mediated during embryonic life, and lymphocytes are continuously produced, it would be necessary that lymphocytes under development during, for example, adult life, had a "memory" of self during the embryonic life, which cannot be claimed to happen. Therefore, while it is well recognized that negative selection does occur, its role as the only mechanism to ensure self-tolerance is inappropriate. It is usually referred to as a recessive mechanism (in analogy to genetics), because it explains self-tolerance based on the absence of self-reactive cells. In contrast, to ensure self-tolerance, it is necessary that additional, dominant, mechanisms act on the system to prevent the activation of the cells not eliminated by negative selection [11].

Recently, a lot of attention has been directed towards regulatory T cells (Tregs), whose roles in the control of the immune system have been extensively analyzed [12,13,14]. One of the reasons for the interest in these cells is that they actively regulate the activation of self-reactive cells, preventing the occurrence of autoimmune diseases. In addition, these cells play an important role in tumor immunity, recognizing tumor-associated antigens and preventing the elimination of tumors, and are under investigation as possible targets of clinical therapies [14]. As reviewed by von Boehmer [15], these cells exert their functions through the secretion of suppressing cytokines (soluble factors that allow for "communication" between cells), such as IL-10 (bystander suppression), and direct cell contact during conjugation (i.e. interaction) with an APC (referred to as linked suppression [16]). An interesting aspect is that the TCR (T cell receptor) repertoire of these cells is highly diverse, and more shifted towards self-reactivity than effector T cells, indicating that Tregs are exported to the periphery in a primed state [17]. The role of Tregs has gained importance with the introduction of experimental systems where the manipulation of animals would lead to the induction of autoimmunity or transplantation tolerance. One such model has been studied by Sakaguchi and colleagues [18,13], where autoimmune diseases were induced in lymphopenic rodents (i.e. with very low numbers of T cells) by transferring a population of $CD25^{-}CD4^{+}$ T cells. In contrast, transferring $CD25^{+}CD4^{+}$ T cells or equal numbers of both cell types would establish self-tolerance. It is important to emphasize that, as commented by León [9], autoimmunity would be induced upon the transfer in the absence of danger signals [19], posing a problem for recessive tolerance-based theories. Recently, it was found that these cells, responsible for the maintenance of self-tolerance, can be identified by the expression of the *Foxp3* intracellular marker [20].

Another model has been investigated by Le Douarin, Coutinho and co-workers, as reviewed in [21,11]. Embryonic tissues transplanted from quails to chick embryos, with approximately the same age, would be rejected once immunological competence had been acquired. This observation went in clear disagreement with the beliefs at the time of these experiments, as to the establishment of natural tolerance during the embryonic period. Using this same experimental system, it was demonstrated that transplantation of the thymic epithelium (TE) would establish tolerance to grafts of peripheral tissues from the same donor, even if tissue-specific antigens were not expressed by the TE. It was demonstrated that donors contained T cells that could lead to the rejection of the graft, but were under the "control" of regulatory cells, specifically selected by the TE. Later on, they have shown that these regulatory cells were capable of "educating", along with tissue-specific antigens, peripheral T cells, which would be induced to express a regulatory phenotype [22]. This process of natural Tregs (that is, those that are continuously exported from the thymus with a regulatory phenotype, e.g. [17]) converting peripheral effector T cells into the expression of regulatory function (induced Tregs) would be later denominated infectious tolerance [23]. Based on these results, a model [22] for the thymus-dependent induction of tolerance to central and peripheral antigens was formulated, where the thymocytes (T lymphocytes under development) would be induced to a regulatory or effector phenotype depending on the avidity of interactions experienced during their development. In that model, the fate of a naïve T cell recognizing an antigen on a peripheral APC would depend on the status of the cell (if it was recently exported to the periphery or if it has been residing in the periphery for some time) and if this APC is concomitantly recognized by a Treg.

3 Modelling Studies Considering Regulatory T Cells

In parallel to the increasing number of experimental systems developed for studying regulatory T cells (Tregs), there has been several modelling studies directed towards such cells. These studies try, through the introduction of a mathematical formalism, to gain some insight into how these cells could be involved in some aspects of the immune system. In this section, some of these models are summarized. It is important to emphasize that the objective here is not to conduct a detailed review of all such studies, but just to highlight some of the main assumptions and outcomes of each model.

León et al. [24] have proposed a general mathematical model for the suppression of effector T cells by regulatory T cells during conjugation with APCs. Three particular models have been studied, and it was concluded that the most plausible suppression mechanism would be based on the active inhibition of effector T cell growth, where the maintenance of a regulatory population would be dependent on the effector population. Later on, the model was extended to account for clonal diversity [25], represented by several sub-populations of regulatory and effector cells. In that model, it was suggested that positive selection would ensure the generation of antigen-specific regulatory T cells, while negative selection would prevent the exporting of too efficient regulatory cells that could bar immune responses to foreign antigens. In addition, the possibility of regulatory T cells suppressing a response directed against foreign ligands, due to their co-presentation with self peptides in APCs, was discussed. In this scenario, the initiation of a response would require the displacement of these self peptides from the antigens or an increase in APC density. Carneiro et al. [26] have analyzed a modification of the model proposed in [24] when confronted with the Sakaguchi [13] experimental system (see section 2). It was verified that, although the new model would account for the experimental results, it would be more robust to a fast increase in APC density in terms of inducing autoimmunity, which is not in accordance with evidence for some autoimmune diseases.

Burroughs et al. [27] have proposed a model to study the consequences of Tregs inhibiting the secretion of IL-2 by effector T cells. Upon stimulation, effector T cells secrete IL-2 and divide in the presence of this cytokine, while Tregs also proliferate in the presence of IL-2, although at lower rates than effector cells, but are not capable of secreting it. In addition, activated Tregs suppress the secretion of IL-2, inhibiting T cell growth. In that model, the density of Tregs would set a threshold on the number of effector cells that have to be activated (thereby secreting IL-2) so that a response could be initiated. Guzella et al. [28] have followed a similar line of investigation. An initial version of a model was presented, aimed at studying the hypothesis that Tregs could control the magnitude of an immune response (to prevent excessive damage to host tissues). In contrast to the work by Burroughs et al. [27], this control would be possible through the secretion of IL-10, in response to the secretion of IFN-γ by effector T cells. It was concluded that it could be valid, although more work needs to be done to verify, e.g., the importance of cytokine diffusion, and the relative numbers of regulatory and effector cells. Although the two models are similar, it is difficult to compare these two studies, due to the absence of more conclusive results in the case of [28].

Kim et al. [29] have proposed a detailed model based on the immune network [30] to study the response of CD8+ (cytotoxic) T cells, taking into consideration the regulation of a response by Tregs. The model considered that effector T cells and Tregs would secrete positive (such as IL-2) and negative cytokines (IL-10 or TGF-β), respectively. In addition, the regulatory capacity of Tregs was modelled as a cell contact dependent mechanism for T cells, which could also render APCs tolerogenic. It was verified that responses in the model were characterized by two phases. Initially, CD8+ T cells would be stimulated in the lymph nodes to rapidly expand. In the second phase, Tregs would proliferate and suppress the CD8+ T cells from further expansion, leading to their migration to the infected tissue. In that model, Tregs would coordinate the response to foreign antigens by inducing the migration of CD8+ T cells from the lymph node to the target tissue.

Carneiro et al. [31] have recently proposed the Crossregulation model, on an effort to incorporate regulatory T cells in a coherent view of the immune system. In that model, it is assumed that to establish self tolerance, it is sufficient to prevent the expansion of autoreactive T cells, during the formation of multicellular conjugates between Tregs and effector T cells in APCs. An important prediction of the model is the partitioning of the peripheral repertoire into three subsets of lymphocytes, related to the density of APCs presenting cognate antigens. The first subset would be composed of lymphocytes which persist for a short amount of time, due to inability to sustain frequent conjugations with APCs. A second subset would be composed of effector T cells whose proliferation is limited by the availability of APCs presenting specific antigens, where the maintenance of Tregs is not sustained, and a third subset, where the APCs can sustain both Tregs and effector T cells.

Finally, using an extension of the model proposed in [24], León et al. [32] have studied the role of Tregs on the development of tumors. The model was based on the assumption that tumor growth induces the presentation of tumor antigens in the lymph node close to the tumor, which would stimulate the growth and the migration of effector and Tregs to the tumor site and initiate tumor elimination. Two tumor growth modes were predicted. In the first mode, characterized by fast growth rate, low immunogenicity and high resistance to the action of effector T cells, it would allow effector T cells to escape the control by Tregs, but could not be eliminated. In the second mode, where the tumor has a low growth rate, intermediate-high immunogenicity and low resistance to effector functions, it induces a balanced expansion of both cell types, with Tregs acting to prevent tumor rejection.

4 Artificial Immune Systems and Regulatory T Cells

At the time of writing and to the knowledge of the authors, the only work in the area of AISs that incorporates Tregs is the Multilevel Immune Learning Algorithm (MILA), proposed by Dasgupta et al. [33], aimed at anomaly detection problems, and that includes also helper T cells, B cells and APCs. In that model, Tregs are generated during a training phase, using self antigens (legitimate or normal data). During the recognition phase, the activation of both B and helper T cells, with no Treg being activated, will lead to the classification of an antigen as nonself, and as self otherwise. Although that work is an important initial attempt to incorporate Tregs into a computational algorithm, the approach followed is, in the opinion of the authors of the present work, too limited, as it does not include a proper interaction between Tregs and the remaining lymphocytes, simply supposing that the former have infinite suppressive capabilities. It is exactly the possibility of adding a form of interaction between the cell types, leading to dynamic computational system that better resembles the natural immune system, that motivates the present work to propose the incorporation of regulatory cells in AISs. It is expected that, in the case of an algorithm aimed at anomaly detection problems (such as MILA) incorporating Tregs can lead to a reduction in false positive rates, considered as analogous to autoimmunity in the natural immune system, given the extensive involvement of these cells in mediating self-tolerance.

Recently, there has been a lot of interest in using additional theories and concepts from Immunology to the development and improvement of AISs. Aickelin et al. [34] have demonstrated particular interest in the application of the Danger Model [19] to the development of intrusion detection systems. So far, that work has lead to the proposal of a new and promising approach, the Dendritic Cell Algorithm (DCA) [35,36]. Twycross and Aickelin [37] have discussed the incorporation of innate immunity, another area in Immunology that has been the target of intense research, into AISs. In addition, Andrews and Timmis [38] point out the disagreement between immunologists in some aspects of the immune system, and discuss the inspiration for incorporating Cohen's model [39]. In this line of research, Stepney et al. [40] have proposed a conceptual framework for bio-inspired algorithms, focused on supporting the development of more biologically plausible computational models. It is important to notice that, in that framework, the construction of models (abstract representations) of the biological system plays a major role. In this sense, the availability of several theoretical models, briefly reviewed in section 3, constitutes a natural source of inspiration in the context of Tregs.

While a possible approach for describing the inclusion of Tregs in AISs would involve identifying characteristics of some application areas that could be related to these cells, in this paper, a different procedure is followed. In order to consider Tregs in AISs, a similar approach to that of Twycross and Aickelin [37] (which have also based the description on the conceptual framework by Stepney et al. [40]) is taken, by describing two simple abstract models that incorporate Tregs and are under investigation by the authors . These models are based on two particular modes of action of these cells, namely through the secretion of cytokines (bystander suppression) and during the formation of conjugates with APCs (linked suppression), as discussed in section 2, and are presented in the following subsections. Obviously, these abstractions are neither unique or definitive, and some of the discussed features may have to be reviewed in the future. Both models consider a continuous source of both effector and regulatory

T cells, and also the eventual death of some cells dependent on some criterion. It must be emphasized that these two models are described with the purpose of providing a general view of possible roles that could be played by these cells in a computational system. It is assumed that specific aspects would be modelled according to some user-selected criteria, and, therefore, their proper incorporation into an specific algorithm will require the consideration of application-specific knowledge. This comes in agreement with a recent work by Freitas and Timmis [4], that argue on the importance of designing an AIS with the target application in mind.

4.1 Bystander Suppression Model

In this model, illustrated in figure 1, the inspiration is the nonspecific suppression of a population composed of n_E effector cells (labelled as $E_1, E_2, \ldots, E_{n_E}$) by n_R Tregs ($R_1, R_2, \ldots, R_{n_R}$), mediated through the secretion of cytokines. It is assumed that an antigenic stimuli (such as the affinity between the cells and antigenic features) is supplied to these two populations. In response to this stimuli, each cell type will secrete a given amount of a typical cytokine, which, for the sake of simplicity, can be characterized as regulatory (secreted by Tregs) or effector (secreted by effector T cells). After this step, the total amounts of secreted cytokines would influence, in a user-defined way, the activation or suppression of each cell type. In this case, if regulatory cells are able to dominate the process, the suppression is nonspecific because it is directed towards all effector cells, independently of the stimuli enlicited in each one of them by the antigen. However, in general, the net influence on each cell would be given by a combination of the an individual factor (the affinity for the antigenic features) and a global response (the total amount of secreted cytokines). In addition, the amounts of cytokines previously secreted should be kept on the system, after the application of a decay term, so that sufficiently strong repeated stimuli could allow effector T cells to escape suppression and initiate a response.

4.2 Linked Suppression Model

In a linked suppression-based model, shown in figure 2, the interplay between effector and regulatory effects during the conjugation of cells to n_A APCs (denoted as $A_1, A_2, \ldots, A_{n_A}$ in figure 2) is considered, where each conjugate can accommodate up to a certain number of cells. When conjugated to an APC, cells would receive stimuli due to the recognition of antigenic features presented in that APC. This model can be considered as a computational implementation of a simplified version of Carneiro's Crossregulation model [31].

In this approach, a set of antigens is distributed throughout the APCs. The conjugation between APCs and T cells will induce a partitioning of the effector and regulatory sub-populations, depending on which cells are taking part in each conjugate at a given time. After each conjugate is established, effector cells can be induced to certain states, such as death, suppression or conversion to a regulatory phenotype, as indicated by the double-ended arrows, depending on

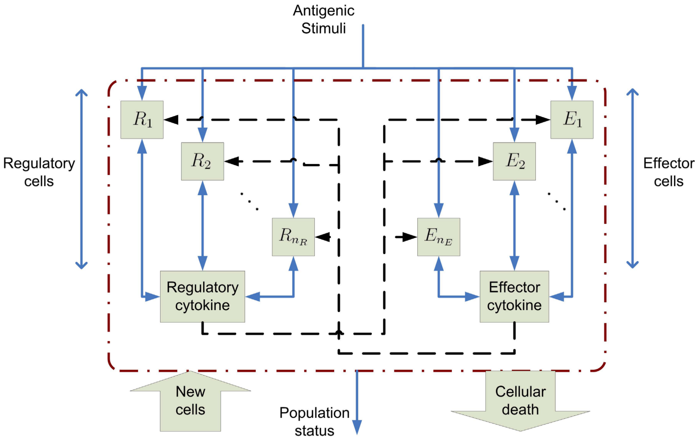


Fig. 1. Bystander suppression-based model

certain conjugate-specific conditions (e.g. the number of Tregs in the conjugate). For the sake of generality, it is assumed that Tregs can be also influenced by the outcomes of each conjugation (leading to, e.g., proliferation or death), and that each cell may take part in more that one conjugate at a time. Finally, the response of the system could be asserted by the population conditions (e.g. relative number of effector and regulatory cells) or the status of each conjugate formed (e.g. in how many conjugates have effector cells proliferated).

In this model, one should notice the introduction of a certain specificity in Treg functions, which can be only targeted at effector cells conjugated in the same APCs. Finally, it can be observed that, if the concentrations of previously secreted cytokines are not considered, the previously discussed bystander suppression model can be considered as a special case of this model, when $A_{n_A} = 1$ and the APC supports a very large number of cells participating in the conjugation.

5 Conclusions

In this paper, some of the basic aspects regarding regulatory T cells have been highlighted. In contrast to the importance of these cells in Immunological research, especially in the last ten years, few immune inspired algorithms incorporate these cells. In view of recent suggestions of incorporating more biologically plausible models in the area of artificial immune systems (e.g. the conceptual model proposed by Stepney et al. [40]), possible starting models for the incorporation of these cells in some algorithms were suggested. In the case of the Linked suppression-based model (section 4.2), it is interesting to notice that it resembles a network-based learning structure, such as a neural network [41], where the connections between processing units (APCs and T cells) are dynamically constructed, depending on the criterion used for generating the conjugations (interactions between APCs and T cells).

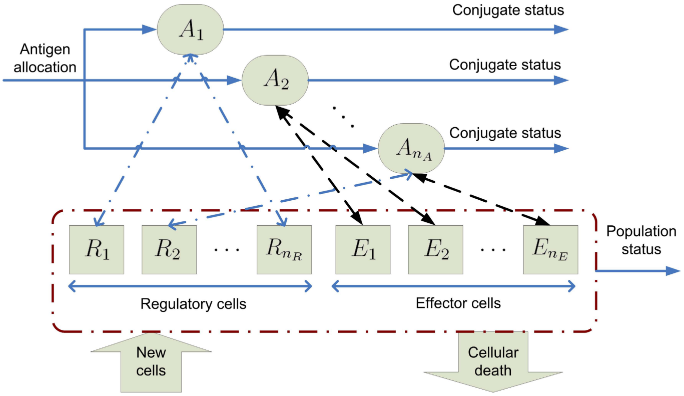


Fig. 2. Linked suppression-based model

It is expected that the incorporation of these cells in some algorithms can not only lead to more biologically plausible models, but also to better results obtained when applying these models to real-life problems, given the importance of these cells in the natural immune system. Future work will be conducted towards implementing initial versions of the Bystander Suppression and Linked Suppression models discussed in sections 4.1 and 4.2 aimed at specific applications.

Acknowledgments. The authors would like to thank the financial support provided by UOL (www.uol.com.br, process number 20060519110414a), FAPEMIG and CNPq (process number 307178/2004-8).

References

1. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach, 1st edn. Springer, London (2002)
2. Garrett, S.M.: How do we evaluate artificial immune systems? Evol. Comput. 13(2), 145–178 (2005)
3. 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, pp. 483–498. Springer, Heidelberg (2005)
4. Freitas, A.A., Timmis, J.: Revisiting the foundations of artificial immune systems for data mining. IEEE Trans. Evol. Comput. (in press) DOI: 10.1109 /TEVC.2006.884042 XXX (2006) XXX
5. Dasgupta, D.: Advances in artificial immune systems. IEEE Comp. Intel Mag. 1(4), 40–49 (2006)
6. Janeway, C.A., Travers, P., Walport, M., Shlonmchik, M.: Immunobiology: the immune system in health and disease, 5th edn. Garland Publishing, Inc. New York, USA (2002)

7. Burnet, F.M.: The clonal selection theory of acquired immunity. Cambridge Press, Cambridge (1959)
8. Schwartz, R.H.: T cell anergy. Annu. Rev. Immunol. 21, 305–334 (2003)
9. León, K.: A Quantitative Approach to Dominant Tolerance. PhD thesis, Universidade do Porto (2002)
10. Grossman, Z., Paul, W.E.: Adaptive cellular interactions in the immune system: The tunable activation threshold and the significance of subthreshold responses. PNAS 89, 10365–10369 (1992)
11. Coutinho, A.: The Le Douarin phenomenon: a shift in the paradigm of developmental self-tolerance. Int. J. Dev. Biol. 49, 131–136 (2005)
12. Coutinho, A., Hori, S., Carvalho, T., Caramalho, I., Demengeot, J.: Regulatory T cells: the physiology of autoreactivity in dominant tolerance and "quality control" of immune responses. Immunol. Rev. 182, 89–98 (2001)
13. Sakaguchi, S., Sakaguchi, N., Shimizu, J., Yamazaki, S., Sakihama, T., Itoh, M., Kuniyasu, Y., Nomura, T., Toda, M., Takahashi, T.: Immunologic tolerance maintained by CD25+CD4+ regulatory T cells: their common role in controlling autoimmunity, tumor immunity and transplantation tolerance. Immunol. Rev. 182, 18–32 (2001)
14. Sakaguchi, S.: Naturally arising CD4+ regulatory T cells for immunologic self-tolerance and negative control of immune responses. Annu. Rev. Immunol. 22, 531–562 (2004)
15. von Boehmer, H.: Mechanisms of suppression by suppressor T cells. Nat. Immunol. 6(4), 338–344 (2005)
16. Davies, J.D., Leong, L.Y., Mellor, A., Cobbold, S.P., Waldmann, H.: T cell suppression in transplantation tolerance through linked recognition. J. Immunol. 156, 3602–3607 (1996)
17. Sakaguchi, S., Ono, M., Setoguchi, R., Yagi, H., Hori, S., Fehervari, Z., Shimizu, J., Takahashi, T., Nomura, T.: Foxp3+CD25+CD4+ natural regulatory T cells in dominant self-tolerance and autoimmune disease. Immunol. Rev. 212, 8–27 (2006)
18. 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(3), 1151–1164 (1995)
19. Matzinger, P.: Tolerance, danger and the extended family. Annu. Rev. Immunol. 12, 991–1045 (1994)
20. Hori, S., Nomura, T., Sakaguchi, S.: Control of regulatory T cell development by the transcription factor Foxp3. Science 299, 1057–1061 (2003)
21. Le Douarin, N., Corbel, C., Bandeira, A., Thomas-Vaslin, V., Modigliani, Y., Coutinho, A., Salaün, J.: Evidence for a thymus-dependent form of tolerance that is not based on elimination or anergy of reactive T cells. Immunol. Rev. 149, 35–53 (1996)
22. Modigliani, Y., Bandeira, A., Coutinho, A.: A model for developmentally acquired thymus-dependent tolerance to central and peripheral antigens. Immunol. Rev. 149, 155–175 (1996)
23. Cobbold, S., Waldmann, H.: Infectious tolerance. Curr. Opin. Immunol. 10, 518–524 (1998)
24. León, K., Peréz, R., Lage, A., Carneiro, J.: Modelling T-cell-mediated suppression dependent on interactions in multicellular conjugates. J. theor. Biol. 207, 231–254 (2000)
25. León, K., Lage, A., Carneiro, J.: Tolerance and immunity in a mathematical model of T-cell mediated suppression. J. theor. Biol. 225, 107–126 (2003)

26. Carneiro, J., Paixão, T., Milutinovic, D., Sousa, J., Leon, K., Gardner, R., Faro, J.: Immunological self-tolerance: Lessons from mathematical modeling. J. Comp. Appl. Math. 184(1), 77–100 (2005)
27. Burroughs, N.J., de Oliveira, B.M.P.M., Pinto, A.A.: Regulatory T cell adjustment of quorum growth thresholds and the control of local immune responses. J. theor. Biol. 241(1), 134–141 (2006)
28. Guzella, T.S., Mota-Santos, T.A., Uchôa, J.Q., Caminhas, W.M.: Modelling the control of an immune response through cytokine signalling. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 9–22. Springer, Heidelberg (2006)
29. Kim, P.S., Lee, P.P., Levy, D.: Modeling regulation mechanisms in the immune system. J. theor. Biol. 246(1), 33–69 (2007)
30. Jerne, N.K.: Towards a network theory of the immune system. Ann. Inst. Pasteur. Imm. 125C, 373–389 (1974)
31. Carneiro, J., Leon, K., Caramalho, Í., van den Dool, C., Gardner, R., Oliveira, V., Bergman, M.-L., Sepúlveda, N., Paixão, T., Faro, J., Demengeot, J.: When three is not a crowd: A crossregulation model of the dynamics and repertoire selection of regulatory CD4+ T cells. Immunol. Rev. 216, 48–68 (2007)
32. León, K., Garcia, K., Carneiro, J., Lage, A.: How regulatory CD25+CD4+ T cells impinge on tumor immunobiology? On the existence of two alternative dynamical classes of tumors. J. theor. Biol. 247(1), 122–137 (2007)
33. Dasgupta, D., Yu, S., Majumdar, N.S.: MILA - multilevel immune learning algorithm and its application to anomaly detection. Soft Comput. 9, 172–184 (2005)
34. Aickelin, U., Bentley, P.J., Cayzer, S., Kim, J., Mcleon, J.: Danger theory: The link between AIS and IDS? Lect. Notes Comput. Sc. 2787, 147–155 (2003)
35. 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)
36. Greensmith, J., Twycross, J., Aickelin, U.: Dendritic cells for anomaly detection. In: Proc. of the IEEE CEC, pp. 664–671. IEEE Computer Society Press, Los Alamitos (2006)
37. Twycross, J., Aickelin, U.: Towards a conceptual framework for innate immunity. Lect. Notes Comput. Sc. 3627, 112–125 (2005)
38. Andrews, P.S., Timmis, J.: Inspiration for the next generation of artificial immune systems. Lect. Notes Comput. Sc. 3627, 126–138 (2005)
39. Cohen, I.R.: Tending Adam's Garden. Elsevier Academic Press, Amsterdam (2004)
40. 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)
41. Haykin, S.: Neural Networks - A Comprehensive Foundation, 2nd edn. Prentice-Hall, Englewood Cliffs (1998)

Automated Blog Design System with a Population-Based Artificial Immune Algorithm*

Kiryong Ha[1], Inho Park[2], Jeonwoo Lee[1], and Doheon Lee[2,]*

[1] Electronics and Telecommunications Research Institute, 305-700, Gajeong dong,
Yuseong-gu Deajeon, Republic of Korea
lyonha@etri.re.kr, ljwoo@etri.re.kr
[2] Dept. of Bio and Brain Engineering, KAIST, 373-1 Guseoung dong, Yuseong-gu,
Daejeon, Republic of Korea
ihpark@biosoft.kaist.ac.kr, dhlee@biosoft.kaist.ac.kr

Abstract. Advances in the Web have eventually arrived at the new concept of 'Web 2.0' and the Blog is a representing service of Web 2.0. Despite the dramatic increase of the Blog users and distinctive characteristics of them, the classical processes of blog creation have difficulties in taking user's preference into account without knowledge about Web programming. Thus, we developed an automated blog design generation system through a population-based artificial immune algorithm. In the algorithm, a user's requirements and a blog design correspond to, respectively, an antigen and an antibody of vertebrate immune system. A slicing tree layout and the HSV color space model are used to represent a blog design as a string format of an antibody. Design quality quantification rules of a blog design and a distance measure between two different blog designs are devised to compose an affinity function. The system shows ability to provide new blogs to the user quickly and easily considering user's preferences with good algorithmic performance when it compared with conventional genetic algorithms.

1 Introduction

The World-Wide Web (W3) has been successfully developed as a pool of human knowledge, in results many of our daily activities are closely related to the Web. These continual improvements of the Web eventually arrived at the new concept of "Web 2.0," whereas its exact meaning remains open to debate. Briefly, Web 2.0 is a conceptual platform that originally comes to guide common characteristics of successful internet services such as Google, Wikipedia and Blog [1]. Even though it does not have a hard boundary, as many emerging concepts, principal keywords of it can be condensed into involvement and collaboration [2].

* This work was supported by Korea Ministry of Information and Communication under Grant C109006020001. KH and JL were partly supported by the IT R&D program of MIC/IITA [2005-S069-02]. We would like to thank CHUNG Moon Soul Center for BioInformation and BioElectronics for providing research facilities.

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 324–335, 2007.

The term ”Blog” is derived from ”Web log.” It generally refers to websites where entries are made in journal style and displayed in a reverse chronological order providing the ability of leaving comments and RSS. Along with basic roles as personal diary and daily opinion column, Blog is regarded as a personal media and a basic tool of personal publishing which will possibly lead a media, communication and online-communities. On that account, Blog is considered as a fundamental service expressing author's individualities. In spite of dramatic increase of Blog users and those characteristics of a Blog service, however, management and manipulation of a Blog page or design is not easy for common users. To have own design, the users have to know how HTML pages are composed and how they are written with programming languages. Therefore, a great number of users rely on Blog hosting services provided by major portal companies. However, such Blogs are maintained without any considerations about the fact that a Blog represents the user's individualities. Namely, Blogs hosted by major companies have limit number of format and those fixed number of designs cannot reflect the user's preferences and desires. Accordingly, many Blogs have similar designs, and some users who want to have original designs have to spend time learning design skills or hiring professional designers. Hence we suggest an automated system for generating Blog design.

Constructing automated Web page design, including blog service, has not been studied a lot except a system that performs automated design on a limited number of elements [3]. The main purpose of their work was to generate style sheet file (CSS) for modifying HTML pages. They focused on finding ”nice” web page and they decided to leave the decision to the user instead of mathematically well defined objective functions because, ”nice” is not quantitative measure, literally. The process is performed by Interactive Genetic Algorithm (IGA). The system starts with randomly generated individuals as genetic algorithm does and the results are shown to the user for selection. The features of the selected individuals are taken into account to compute next generation. Newly generated individual are presented again, and this process repeated until the user satisfies with results. They dealt with elements related to a general style of pages, such as background color, rules, bullets and arrows as well as paragraph and text style elements. Although users may actively retrieve a new design throughout the system, it can alter only limited elements rather than overall layout and composition of a page. In other words, the user is only capable of altering minor aspects of the design. Furthermore, the population size is limited because of all individuals have to be presented and evaluated by the user, requiring a long time until a workable design is obtained.

This paper aims to propose an automated system for generating Blog designs based on a population-based artificial immune algorithm. Section 2 describes the problem with system requirements to overcome limitations of previous approaches. Section 3 shows the system architecture and structure with a whole work flow diagram. It also provides examples of Blog designs generated by the proposed system. Section 4 discusses the results, and conclusion will be followed in Section 5.

2 Problem Descriptions

Most of classical processes of blog creation have difficulties in taking user's preference into account without knowledge about Web programming. Thus, our system focuses on both at the same time; diminishing user intervention and reflecting user preference. Also, we intend to generate overall blog designs including layout and component composition not like previous researches worked on limited number of attributes.

To achieve these requirements, we regard the design generation process as an optimization problem of finding best Blog designs by optimizing multiple criteria to gratify users. For this purpose, we applied AIS algorithm based upon the metaphors of the natural immune system [10]. AIS has been attempted to exploit theories, principles, and concepts of modern immunology to solve problems in science and engineering. Indeed, immunological theories had been applied to produce solutions for complex problem, including optimization [11]. The majority of publications for optimizations are based on the clonal selection principle, resulting in a number of algorithms such as the Clonalg [12], op-AINET [13] and B-Cell algorithms [14]. Even though it is certainly true that many of clonal selection principle based algorithms are familiar with evolutionary algorithms (EA), diversity mechanisms and a memory mechanism are considered as distinctive features [11].

Memory mechanism (Reinforcement learning and memory). In wake of first (primary response) infection involving a particular antigen, the responding naive lymphocytes proliferate to produce a colony of cells, and some persist as the memory cells that can survive for years, or indeed even for a lifetime. Memory cells circulate through the blood, lymph and tissues and when exposed to an antigen (secondary response), it starts to differentiate and produce high affinity antibodies. That, using previously remembered antibody rather than 'starting from the scratch' every time ensures both the speed and accuracy of the immune response.

Diversity mechanisms (Somatic hypermutation, receptor editing). The repertoire of antigen-activated B cell is diversified basically by two mechanisms, hypermutation and receptor editing [15]. Hypermutation is mutating individuals with proportional to the affinity and this strategy will lead the algorithm to perform a greedy search (exploitation of the surrounding space) [12] [16]. Receptor editing offers the ability to delete their low affinity receptors and developed entirely new ones through V(D)J recombination [17]. It introduces diversity, leading to possibly better candidate receptors even though the start affinity might be lower.

Through the adoption of the memory mechanism and diversity mechanisms, suggested system behaves dynamically to increase its ability for generating (optimizing) Blog designs. Table 1 shows simple analogy of design generation process with artificial immune algorithms. A user's preference and a blog design correspond to, respectively, an antigen and an antibody of AIS framework. Accordingly, affinity is considered as fitness between the design result and user's

Table 1. Analogy with artificial immune system

Antibody	Blog design(layout and color bit string)
Antigen	User preference
Affinity	Design evaluation function

satisfaction. Here, we utilized the Clonalg algorithm [12] with our own antibody encoding scheme. Accordingly, affinity of an antibody is calculated based on design evaluation measurement which will be explained in next section.

3 System

3.1 Antibody Representation (Blog Design Representation)

In population-based AIS algorithms, a population is defined as a set of antibodies which are candidate solutions. Here, a solution is a blog design which satisfies given criteria. Therefore, an antibody should contain all the information about blog design. To design an antibody, we consider a blog as independent functional modules and a set of decorations for each modules. We divided a blog as several components based on its functional ability and characteristics of contents. Figure 1-a) shows 12 components of a Blog and Figure 2 shows decorative colors respectively. According to this categorization, an antibody can be represented by a layout of functional modules and their decorative elements (colors).

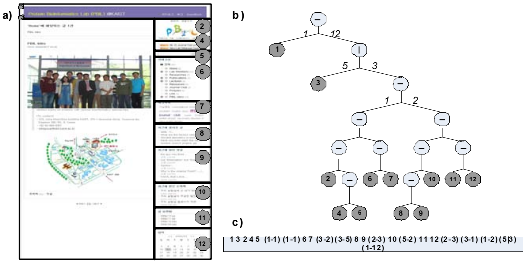


Fig. 1. a) Functional modules of Blog, b) Slicing tree representation, c) Equivalent string representation

Layout. To encode a blog layout, arbitrary size and location of every component in a blog should be represented in a formal way which should be easily and efficiently handled in AIS. Slicing tree representation, commonly used in the area of facility layout, is successfully used to represents a blog layout as a string format by recursive divisions of a given page. For example, it recognizes layouts of figure 3-a) as a partitioning process [19]. Horizontal and vertical partitions are represented as slicing trees as figure 3-b) and trees are equivalently converted to a string based on postfix tree traversal. Tree traversal string has two types

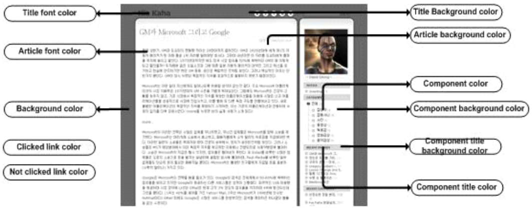


Fig. 2. Color composition of a Blog design

of elements; one is operand for leaf node in tree, the other is operator for intermediate node in tree. Operand corresponds to ID of given area (component identifier) and operator express the way of how given area is divided - indicating direction (horizontal or vertical) with a division ratio.

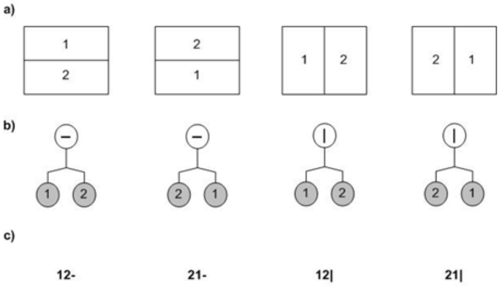


Fig. 3. a) Horizontal and vertical division of given page, b) Tree format of slicing tree representation

Color. Color is another main feature of blog design. Designs are affected by the colors used in each element as well as the way of their combinations. Among 3-channel color spaces, such as RGB, HSV, YUV, YIQ, CIE LAB, and CIE LUV [5], we used HSV color space known to a perceptual color space and most frequently used color space in color comparison. That is, the three components H (Hue), S (Saturation) and V (Value) correspond to the color attributes closely associated with the way that the human eyes perceive the colors [7].

3.2 Affinity

Affinity is a degree of binding between a cell receptor and a antibody (higher affinity, stronger binding) and it is a standard method to evaluate fitness of each design with given criteria. We made up affinity function as weighted sum of three sub-functions corresponding to system requirements.

Design Evaluation. We classify principal calculation methods of sub-functions into two types of measurements; quantification of a certain design and distance between two designs. One is a trail to quantify the quality of given design and the other is focusing on calculating difference between two different designs.

Design quality quantification. It is aimed to inspect basic web page design rules for a given page to be a well-formed Blog design. We set up basic rules based on web page design criteria and see how a given antibody fits. Total about of value is calculated as below with normalized.

$$Rule(bd_i) = \sum_{j=1}^{3} Rule_j \quad (bd_i \text{ is an arbitary blog design, } 0 < Rule_j < 1) \qquad (1)$$

1. Preferred size($Rule_1$)
   Check the aptness of the size and position of every component. We compare the size of generated components to common range of predefined size. Distance is defined as amount of deviated size from given range.
2. Readability($Rule_2$)
   Examine readability of the article. This can be done by inspecting difference of brightness and saturation level between a letter color and a corresponding background color.
3. Color combination($Rule_3$)
   Calculate average hue distance of all backgrounds color to analyze the ranges of the average distance. We expect the average hue distance is either short for harmonic or long for emphasizing.

Distance between two designs. Along with quantification of a single page, calculating distance between two different designs is another fundamental measurement. We calculate distance by comparing layout and color of the design.

$$D(bd_i, bd_j) = w_1 \times D_{layout}(bd_i, bdj) + w_2 \times D_{color}(bd_i, bd_j) \qquad (2)$$

1. Layout distance (D_{layout})
   we should analyze both macroscopic and microscopic difference at the same time while measuring a distance of two layouts. To see microscopic difference (D_{micro}), we simply compare size and location of every component. In case of macroscopic difference (D_{macro}), however, we exploit characteristics of slicing tree layout representation. As mentioned above, it expresses layout as a recursive partitioning of a given page, which means that given page is partitioned into horizontal or vertical recursively from the big piece. Thus macroscopic view of give page can be achieved by regulating (limiting) depth of recursive partitioning. Figure 4 shows an example. Figure 4-a) originally represents layout 4-b) and an overall view of layout can be seen as figure 4-c) by limiting number of tree depth (gray box). Based on this, macroscopic distance is calculated by comparing size and location of each corresponding parts.
2. Color distance (D_{color})
   Color histogram is one of the most widely used scheme and it captures the global color distribution of an image [8]. It is also known as one of the fastest (simplest) method and contains no spatial information, as we intended (we already consider spatial information in D_{layout}). We take the quantization of

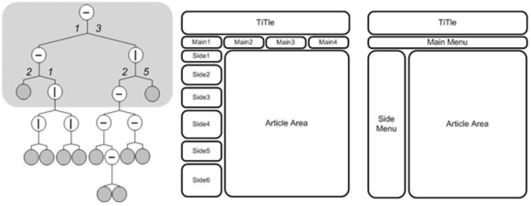


Fig. 4. a) Slicing tree representation, b) corresponding detail layout, c) macroscopic layout (gray box area)

Table 2. Category of color

Color category	Huge-range	Saturation-range	Value-range
White	Any	< 20	≥ 85
Black	Any	Any	< 25
Gray	Any	< 20	[25, 85)
Red	[350, 25)	20	25
Red-Yellow	[25, 45)	20	25
Yellow	[45, 65)	20	25
Yellow-Green	[65, 85)	20	25
Green	[85, 160)	20	25
Green-Blue	[160, 180)	20	25
Blue	[180, 270)	20	25
Blue-Purple	[270, 290)	20	25
Purple	[290, 330)	20	25
Purple-Red	[330, 350)	20	25

the bin considering all three color channels together in [9], because the computational speed of color histogram largely depends on the size of binning. Every pixel of Blog is categorized based on their H, S and V values as table 2. Color distance is measured by the Euclidean distance of histogram.

Affinity function. We compose affinity function based on above two design evaluation measurements. Table 3 illustrates affinity sub-functions and their purpose. These sub-functions are aimed to achieve system requirements - diminishing user intervention and reflecting user preference.

1. Self evaluation affinity
   It is usually performed by the user or web page designer by 'trial and error'. Here we intend to reduce user's task by checking up the web page design rules automatically. Design quality quantification measurement exactly matched to this type of affinity function.
2. User preference affinity
   User preference affinity is devised for affecting user's personal preference into automated process. For that, system let the user select his (or her)

Table 3. Affinity functions and their definitions

Affinity function	Purpose	Definition
Self Evaluation	Reduce user intervention by guaranteeing minimum design quality	Rule-based quantification
User Preference	Reflect user preference by inspecting similarity with user's selection	Distance with user selected design
Popular Set	Consider popular blog design by calculating distance	Minimum distance among popular blogs

favorite macroscopic layout style and color mood. Then, we compare it with candidate designs.

3. Popular set affinity
   Together with user preference affinity, it is another way of increasing user's satisfaction by considering features of existing popular set of blog. New individuals are affected by the existing blog information and furthermore overall blog set can be changed according to the changes of each individual. Popular blogs are dynamically selected based on number of visitors and these will lead a change.

With three sub-functions, affinity value of a given design (*bd*) is determine by weighted sum of each sub-function. These weights can be controlled by the user in the first step of the generation process.

$$\begin{aligned} Affinity(bd) &= \alpha \times SelfEvaluating + \beta \times UserPreference + \gamma \times PopularSet \\ &= \alpha \times Rule(bd) + \beta \times D(bd, bd_{selected}) \\ &\quad + \gamma \times min_{j \in popular\ blogs}(D(bd, bd_j)) \end{aligned} \tag{3}$$

3.3 Systems

Figure 5 shows overall system flow diagram. Generally system is consists of 3 steps and each step corresponds to system structure; get user information, generate a blog design and return the result to the user.

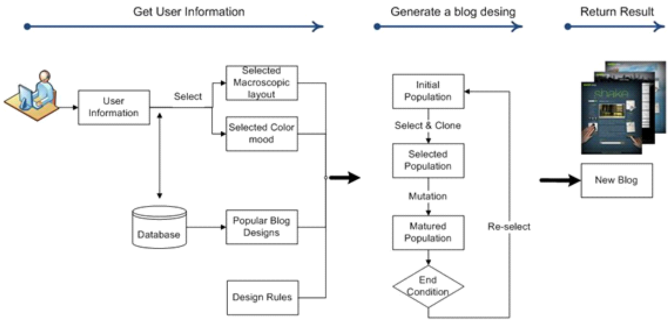


Fig. 5. Flow diagram of overall system

4 Result

Outputs of suggested system are new blogs with generated designs as in Figure 6. It clearly appears that there are various types of color moods and layout formation.

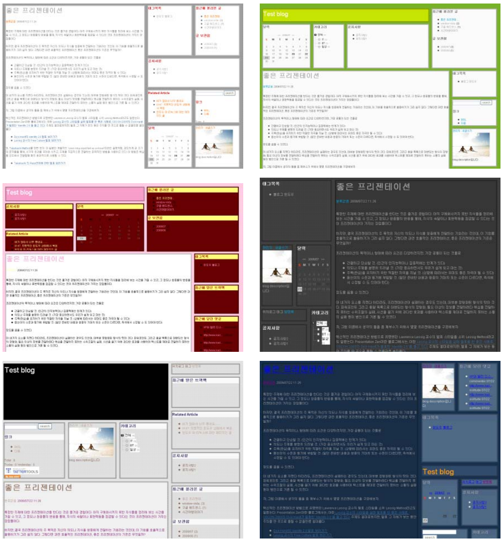


Fig. 6. Examples of generated blog design

In Figure 7-a), changes of affinity values are recorded along evolution number according to population size. In this graph, x-axis represents number of evolution and y-axis is corresponding affinity values. Graph shows that the affinity value converges to a stable state after around 20 number of evolution. Also maximum affinity values become similar above 2560 population size. At this condition (2560 population size and 20 evolutions), indicated by red box, requires only 30 40 seconds (average 37.4 second in 100 runs) and it is fast enough for providing a new blog design to the user. In this problem, we are dealing with subjective problem related to artistic view. It obviously consists of many variables such as layout information and colors information. Furthermore, complex relationships among these variables are built throughout whole generation processes. This condition usually makes a large number of local optimums and we convinced that AIS algorithms can perform better in such situation. In fact, the clonal selection algorithm is known to good at multi-modal optimization by reaching a diverse set of local optima solutions, while the conventional GA tends to polarize the whole population of individuals toward to the best candidate solution [12][18]. In figure 7-b) the comparisons of affinity value between Clonalg and genetic algorithms are performed at the same affinity function (fitness function for genetic algorithms). Y-axis is affinity (fitness) value and it is an average value of 100

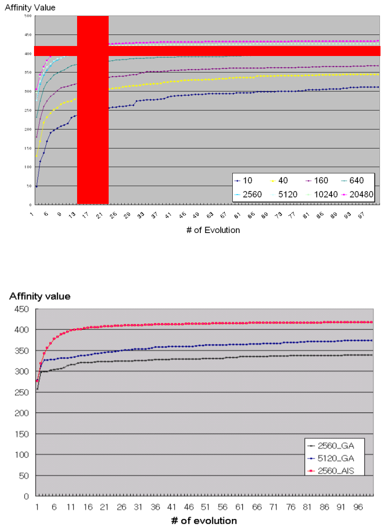


Fig. 7. a) Affinity maturation graph according to number of evolution and population size, b) Affinity graph of AIS and GA at same condition

experiments. Graph clearly shows that AIS algorithm performs better than GA at every evolution number even compared with 5120 population size.

5 Concluding Remarks

An automated blog design system is developed through the adoption of the main feature of vertebrate immune system for solving blog design generation problem. The system exploits artificial immune algorithms at generation process by regarding user preference as an antigen and blog design as an antibody. AIS provide a good metaphor and it also shows good performance. With algorithmic performance, we tried to achieve two competing requirements at the same time - diminishing user intervention and reflecting user preference. User interventions are mainly decreased by AIS that performs automation (optimization) processes and it filtered out unreasonable designs applying design quality quantification, named "self evaluating affinity". Also customization chance is systematically provided by letting the user select macroscopic layout and color moods at the early step of generation and considers the selection in "user preference affinity". We expect to increase user's satisfaction throughout the comparison process with current popular blogs based on "popular set affinity." Suggested system adapted and improved various techniques to represent blog design as an antibody and developed measurements to evaluate blog design quantitatively. Slicing tree layout representation and HSV color space are proposed to represent blog design as a string format as well as design quality quantification rules and distance between two different designs are devised to compose affinity function. At last, the system is capable of quickly providing new blogs to the user and easily considering users preferences. This system is almost new types of application that automatically generates whole web page design and this sort of web applications which are based on artificial intelligence will be studying more and more with a new concept of Web 2.0.

References

1. Davis, I.T.: Web 2.0 and All That, Internet Alchemy Blog (July 4, 2005), http://internetalchemy.org/2005/07/talis-web-20-and-all-that
2. O'Reilly, T.: What is Web 2.0 (September 30, 2005), http://www.oreilly.com/go/web2
3. Monmarch?, N., Nocent, G., Venturini, G., Santini, P.: Imagine: a tool for generating HTML style sheets with aninteractive genetic algorithm based on genes frequencies. Systems, Man, and Cybernetics. In: IEEE SMC '99 Conference Proceedings (1999)
4. Geigel, J., Loui, A.: Automatic Page Layout Using Genetic Algorithms for Electronic Albuming. In: Proceedings of Electronic Imaging (2001)
5. Cheng, H.D., Jiang, X.H., Wang, J.: Color Image Segmentation: Advances and Prospects. Pattern Recognition (2001)
6. Timmis, J., Edmonds, C., Kelsey, J.: Assessing the performance of two immune inspired algorithms and a hybrid genetic algorithm for Function Optimization. In: Evolutionary Computation, CEC2004 (2004)
7. Cheng, H.D., Jiang, X.H., Wang, J.: Color Image Segmentation Based on Homogram Thresholding and Region Merging. Pattern Recognition (2001)
8. Li, X., Chen, S.-C., Shyu, M.-L., Li, S.-T., Furht, B.: A Novel Hierarchical Approach to Image Retrieval Using Color and Spatial Information. In: Proceedings of the Third IEEE Pacific Rim Conference on Multimedia, Advances in Multimedia Information Processing, IEEE Computer Society Press, Los Alamitos
9. Androutsos, D.: Efficient Indexing and Retrieval of Color Image Data Using a Vector-Based Approach. Ph.D Dissertation (1999)
10. Timmis, J.N.: Investigating the Evolution and Stability of a Resource Limited Artificial Immune System. In: Proceeding of the Genetic and Evolutionary Computation Conference, Workshop on Artificial Immune Systems and Their Applications, pp. 40–41 (2000)
11. Hart, E., Timmis, J.: Application Areas of AIS: The Past, The Present and The Future. In: Nicosia, G., Cutello, V., Bentley, P.J., Timmis, J. (eds.) ICARIS 2004. LNCS, vol. 3239, Springer, Heidelberg (2004)
12. De Castro, L., Von Zuben, F.J.: The clonal selection algorithm with engineering applications. In: GECCO 2002, Workshop Proceedings, pp. 37–37 (2000)
13. 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, Idea Group Publishing, USA (2001)
14. Timmis, J., Edmonds, C., Kelsey, J.: Assessing the performance of two immune inspired algorithms and a hybrid genetic algorithm for optmisation. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 308–317. Springer, Heidelberg (2004)
15. George, A.J.T., Gray, D.: Receptor Editing during Affinity Maturation. Imm. Today 20(4), 196 (1999)

16. Sahan1, S., Polat1, K., Kodaz2, H., Gönes, S.: The Medical Applications of Attribute Weighted Artificial Immune System. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, Springer, Heidelberg (2005)
17. Nussenzweig, M.C.: Immune Receptor Editing; Revise and Select. Cell 95, 875–878 (1998)
18. He, Y., Hui, S.C., Lai, E.M.-K.: Automatic Timetabling Using Artificial Immune System. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, Springer, Heidelberg (2005)
19. Cheng, R., Gen, M.: Genetic search for facility layout design under interflows uncertainty. Japanese Journal of Fuzzy Theory and Systems 8, 335–346 (1996)

Immune and Evolutionary Approaches to Software Mutation Testing

Pete May[1], Jon Timmis[2], and Keith Mander[1]

[1] Computing Laboratory, University of Kent, Canterbury, Kent, UK
petesmay@gmail.com, k.c.mander@kent.ac.uk
[2] Departments of Computer Science and Electronics, University of York, York, UK
jtimmis@cs.york.ac.uk

Abstract. We present an Immune Inspired Algorithm, based on CLONALG, for software test data evolution. Generated tests are evaluated using the mutation testing adequacy criteria, and used to direct the search for new tests. The effectiveness of this algorithm is compared against an elitist Genetic Algorithm, with effectiveness measured by the number of mutant executions needed to achieve a specific mutation score. Results indicate that the Immune Inspired Approach is consistently more effective than the Genetic Algorithm, generating higher mutation scoring test sets in less computational expense.

1 Introduction

Software testing can be considered to have two aims [1]. The primary aim is to prevent *bugs* from being introduced into code - prevention being the best medicine. The second is to discover those un-prevented bugs, i.e. to indicate their *symptoms* and allow the *infection* to be *cured*.

Curing an infection is a two stage process of identifying and then correcting faults. These continue until all bugs in the code have been found, at which point a set of tests will have been generated that have reduced the failure rate of the program. Unfortunately, a tester does not know *a priori* whether faults are present in the software, posing an interesting dilemma: *how does a tester distinguish between a "poor" test that is incapable of displaying a fault's symptoms, and a "good" test when there are simply no faults to find?* Neither situation provides a useful metric. A heuristic to help aid this problem uses the notion of *test set adequacy* as a means of measuring how "good" a test set is at testing a program [2]. The key to this is that "goodness" is measured in relation to a predefined *adequacy criteria*, which is usually some indication of program coverage. For example, statement coverage requires that a test set executes every statement in a program at least once. If a test set is found inadequate relative to the criteria (e.g. not all statements are executed at least once), then further tests are required. The aim therefore, is to generate a set of tests that fully exercise the adequacy criteria.

Typical adequacy criteria such as statement coverage and decision testing (exercising all true and false paths through a program) rely on exercising a

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 336–347, 2007.

program with an increasing number of tests in order to improve the reliability of that program. They do not, however, focus on the cause of a program's failure, namely the faults. One criteria does. Known as *mutation testing*, this criteria generates versions of the program containing simple faults and then finds tests to indicate their symptoms. If an adequate test set can be found that reveals the symptoms in all the faulty program versions, then confidence that the program is correct increases. This criterion forms an adequacy measure for the *cure*.

In previous work, we outlined a vision for a software mutation system that exploits Immune-Inspired principles [3]. In this paper we present a limited set of our results from our investigations, detailed further in [4]. The remainder of this paper is organised as follows: Section 2 describes the mutation testing process. Next, in section 3, the notion of algorithm effectiveness with respect to evolving test data using mutation testing is introduced. Section 4 details the Immune and Genetic algorithms, which are compared in section 5.

2 Mutation Testing

Mutation testing is an iterative procedure to improve test data with respect to a program, as indicated in Figure 1. The initial parameters to the process are the *PUT* (Program Under Test), a set of mutation operators (Mutagens), and a test set population, *T*. Initially, by using an oracle, the PUT must be shown to produce the desired outputs when executed with the test set T. If not, then T has already demonstrated that the PUT contains a fault, which should be corrected before resuming the process.

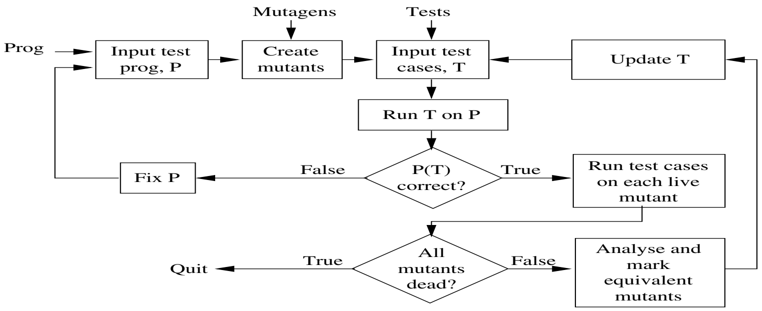


Fig. 1. The Mutation Testing process. *Diagram reproduced from [4], modified from [5].*

The next stage is to generate a set, *M*, of fault induced variants of the PUT that correct for simple faults that could have occurred. Each variant, or *mutant*, differs from the PUT by a small amount, such as a single lexeme, and is generated by a mutagen. These mutation operators alter the semantics of the PUT depending on the faults they classify. For example, the *relational operator* mutagen will generate a number of mutants where each one has an instance of a relational operator replaced by another. Typically, the 22 mutagens developed for Fortran-77 programs (listed in [5]) are used.

The generated mutants are then executed with all tests in T and their outputs compared against the outputs from the PUT. If a mutant produces a different result from the PUT for any test, then the fault corrected by the mutant is proven not to occur in the PUT. Subsequently, the tester's confidence that the PUT is correct, increases. Furthermore, any such undesirable mutants become superfluous, as a test exists to distinguish them, and so they are *killed* (removed) from the mutant set M. Once all the tests in T have been executed on all mutants in M, those mutants that remain alive (that still exist in M) are so far indistinguishable from the original. In other words, there does not exist a test in T that will cause these *living* mutants to compute a different output from P. These mutants become the target for the next iteration, where new test data will be generated in the attempt to detect them. This process continues until all mutants in M are killed. Killing mutants, however, is not a trivial task as some mutants may be semantically the same as the PUT. These mutants are known as *equivalent mutants*, and will always produce the same output as the PUT regardless of the test applied. As a consequence, M can never be completely emptied when equivalent mutants exist. This has an adverse effect on mutation testing as the tester does not know whether the tests that remain in M are equivalent or not. If they are equivalent then no test will kill them; if not, then a test that will distinguish them has so far not been found.

Reducing the set M to the empty set offers a useful metric for assessing the quality of T with respect to the PUT. If T manages to kill all non-equivalent mutants then the tests are capable of identifying that none of the faults the mutants try to repair are present in the PUT. Before T reaches this adequate state however, it will only discriminate a proportion of the mutants, indicated by the number of non-equivalent mutants killed from M. This proportion is the *mutation score* (MS) and is more formally defined as:

$$MS = \frac{|mutants_{killed}|}{|mutants| - |equivalents|} \tag{1}$$

That is, the proportion of mutants killed (identified) out of all non-equivalent mutants. As this proportion increases (i.e. more non-equivalent mutants are killed), so does the adequacy of the test data and the tester's confidence in the correctness of the PUT. Subsequent iterations therefore involve generating new tests to improve the adequacy of T.

3 Evolving Test Data

An initial test set for a program undergoing mutation testing can result in 50-70% of non-equivalent mutants being killed [6]. Improving this figure is the prime motivation for testers to undertake mutation testing's cumbersome manual process. While Genetic Algorithms, amongst other techniques, offer a beneficial reduction to the amount of work a tester has to perform, our research has investigated the use of Artificial Immune Systems to improve these results.

The primary hypothesis of our work is that an Immune Inspired Algorithm for Mutation Testing (IIA/MT) is consistently at least as effective at evolving test data as a Genetic Algorithm for Mutation Testing (GA/MT). This is, in itself, a high level hypothesis that requires explanations of "consistency" and "effectiveness" in this context. *Consistency* simply refers to the notion that whilst an algorithm could be at least as effective for a given program on a given run, this may not be the case over multiple runs. An algorithm must be at least as effective *on average*, in order to be *consistent. Effectiveness* is a loose term, given to indicate some measure of performance an algorithm has in improving test data. Predominantly, the main performance concern with an algorithm is that every iteration, each new test within the population has to be executed against the remaining living mutants in order to judge its strength (in identifying mutants as incorrect). As every mutant executed needs a finite amount of time to execute[1], the more mutants an algorithm executes, the longer it will take to run. Ultimately, the number of mutant executions needed depends largely on the algorithm employed and its representation of the problem space, and therefore is a good measure of an algorithm's *effectiveness.*

As an example, consider the situation where a GA/MT requires 5 million mutant executions to achieve a 95% mutation score for a given program, P. For an IIA/MT to be *at least as effective* for the same program P, requires that a 95% mutation score is achieved in 5 million mutant executions or fewer. Alternately, *at least as effective* can be viewed as the IIA/MT achieving a mutation score greater than or equal to 95% after 5 million executions. Either way, emphasis is placed on obtaining higher mutation scores in fewer mutant executions, or, considering the number of executions relates directly to algorithm run times, as achieving a higher mutation score in less time.

This paper makes a comparison, using the measure of effectiveness specified above, between two evolutionary algorithms used to automatically evolve software test data: an Immune Inspired Algorithm and a Genetic Algorithm. This section outlines the process by which test data is evolved using mutation testing.

3.1 Our Approach to Evolving Test Data

Like all evolutionary algorithms, both the GA/MT and the IIA/MT algorithms we have developed iteratively optimise a population of individuals in an attempt to find a good solution - in this case, a set of tests which kill all mutant programs. The overall process by which an individual is evolved is common to both algorithms, and presented in figure 2. This is based on the mutation testing process outlined in figure 1.

Both algorithms have subtly different ways of evolving the population of tests. Understanding these differences will be useful in explaining any observed differences in the results. The following section describes the two evolutionary

[1] Mutants that enter infinite loops can have their execution times limited by a multiplication of the original program's execution time.

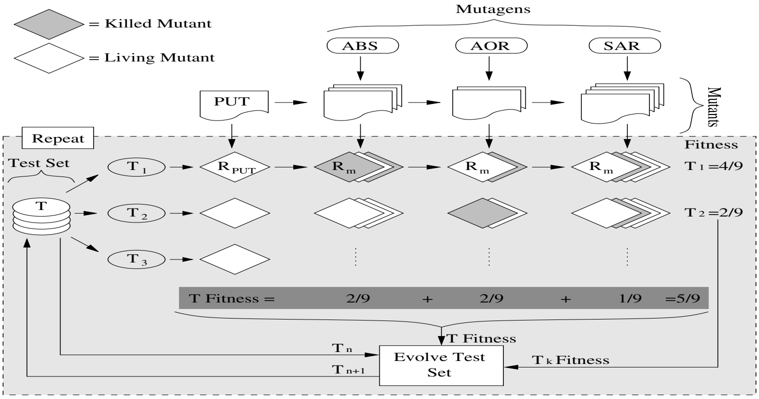


Fig. 2. Test data evolution using mutation testing

algorithms (one genetic, one immune) developed to evolve test data using mutation testing.

4 Immune and Genetic Systems Developed

4.1 Genetic Algorithm for Mutation Testing

Genetic Algorithms are inspired by Darwinian theories of species evolution in response to environmental pressures [7]. In terms of mutation testing, the GA/MT algorithm iteratively evolves a population of individuals in an attempt to kill the most non-equivalent mutants. Each individual is **a set of** m **tests**, for example, [<1,2,3>, <-1,5,7>, ..., <99,42,8>], for the *TriangleSort* program. For any individual, at any iteration, the proportion of non-equivalent mutants killed by all its tests combined is a measure of its **fitness** (**note:** this is not simply the sum of each test's mutation score, as indicated by the "T Fitness" input to the "Evolve Test Set" box in figure 2). Evolution is guided by each individual's fitness - fitter individuals are more likely to survive, and therefore more likely to reproduce than less fit individuals, meaning these solutions (tests) are more likely to be incorporated into the next generation. At the end of this iterative process, the best individual (set of tests) is returned to the tester.

Algorithm 1 shows the pseudocode for the GA/MT algorithm (methods used are described in [4]), which works as follows: Let the population in the i-th iteration be called P_i, initialised to contain s individuals, with m tests each, that are either randomly generated or specified by the tester. n iterations are performed. Every iteration, each individual in P_i has its affinity (or fitness) calculated (*line 6*). Each affinity is then normalised (*line 8*) against the sum of all the affinities (*line 7*) in preparation for roulette wheel selection. Next, the best test is selected and added as the first member of the child population (*lines 9-10*). Further individuals are added to the child population through a three step

```
              $n$          the number of iterations to perform
              $s$          size of main population
    inputs:   $m$          the number of tests in each individual
              $crossRate$  the probability of crossover occurring
              $mutRate$    the probability of a test being mutated
 1  begin
 2      $i \leftarrow 0$
 3      $P_i \leftarrow$ initPop($s$)                    // initialise main population
 4      while $i < n$ do
 5          $Ch \leftarrow \{\}$                       // initialise child population
 6          calculateAffinity($P_i$)
 7          $total \leftarrow$ sumAffinities($P_i$)
 8          normaliseAffinities($P_i$, $total$)
 9          $B \leftarrow$ getBestTest($P_i$)                              // get best Test
10          $Ch \leftarrow$ combine($Ch$, $B$)             // and add to child population
11          while size($Ch$)$< s$ do
12              $I_1 \leftarrow$ rouletteSelection($P_i$)
13              $I_2 \leftarrow$ rouletteSelection($P_i$)          // select 2 Individuals
14              $Q_{1,2} \leftarrow$ singlePointCrossover($I_1$, $I_2$, $crossRate$)    // crossover
15              $Q_1 \leftarrow$ mutateChild($Q_1$, $mutRate$)                   // and mutate
16              $Q_2 \leftarrow$ mutateChild($Q_2$, $mutRate$)
17              $Ch \leftarrow$ combine($Ch$, $Q_{1,2}$)          // add to child population
18          end
19          $P_{i+1} \leftarrow Ch$
20      end
21      return getBestTest($P_i$)    // return the highest fitness individual
22  end
```

Algorithm 1. Genetic Algorithm for Mutation Testing

process of: selecting two parents using roulette wheel selection (higher affinity individuals have a higher probability of being selected - *lines 12-13*); performing single point crossover (the tails of each parent individual, from a randomly chosen point, are swapped - *line 14*); and finally, performing mutation on randomly chosen tests within each crossed-over individual (*lines 15-16*). These children are added to the child population (*line 17*), and the process repeats until it contains s individuals. The child population then becomes the next iteration's parent population, P_{i+1}.

4.2 Immune Inspired Algorithm for Mutation Testing

The Immune Inspired Algorithm for Mutation Testing is based on the Clonal Selection theory. More specifically, the IIA/MT was based on the CLONALG implementation [8], although alterations were made to focus the algorithm on the mutation testing problem domain, in particular removing the concept of a memory individual per antigen, and instead allowing many memory individuals to contribute to an antigen's recognition.

The algorithm iteratively evolves a population of antibodies, where each antibody is **a single test** (e.g. <1,2,3> for the *TriangleSort* program), searching for those which kill at least one mutant program not already killed by an existing memory test. Those antibodies (tests) that are found to be useful are added to the memory set to be returned to the tester at the end of the process. The constraint that an antibody must kill at least one living mutant is an entry requirement to the memory set itself. Instead the **affinity** (or fitness) of an antibody is its mutation score, as calculated by equation 1, allowing tests to be sought that primarily kill the most mutants. Antibody evolution occurs through the process of Clonal Selection, guided by the affinity values; high affinity antibodies generate more clones than low affinity ones, but mutate less (as they are closer to the desired solution).

Pseudocode for the IIA/MT is shown in Algorithm 2 (methods used are described in [4]), which works as follows: Let the population in the i-th iteration be called P_i, initialised to contain s tests, either by randomly generating tests or by specifying each one. The memory set is initially empty. Every iteration, each antibody in P_i has its affinity calculated (i.e. its mutation score) with a record kept of which mutants each antibody kills (*line 6*). Useful antibodies (those that kill at least one mutant not killed by any antibody in the memory set) are added to the memory set, M (*line 7*). `nFittest` highest affinity antibodies are then selected from P_i (*line 8*). These are combined with `nFittest` random[2] antibodies selected from the memory population (*line 9*). In neither case are the selected tests removed from their respective populations. `nFittest` antibodies are randomly chosen from this combined set and, together with the useful antibodies selected earlier, undergo clonal selection (*lines 10-13*). Antibody cloning is proportional to its affinity, although a minimum of 1 clone per parent antibody is created, and all clones undergo mutation inversely proportional to their affinity (mutation score). Useful cloned antibodies are added to the memory set (*line 14*). `nFittest` cloned antibodies are added to P_i, and a number of worst affinity antibodies in P_i are removed until P_i size equals s. Finally, `nWorst` antibodies in P_i are replaced by new, randomly generated ones (*line 15*). This process repeats for n iterations.

4.3 Differences in the Algorithms

Both algorithms evolve a population of individuals in an attempt to find a good solution to the problem they encode. However, a fundamental difference is that, in general, a GA aims to find the best individual to encompass the problem domain as opposed to an IIA which evolves a set of specialist individuals to encompass the problem. This has an impact on an individual's representation.

[2] Antibodies are selected at random from the memory population because a "useful" antibody does not necessarily have a high mutation score (e.g. it may only kill a single mutant). Selecting only high affinity memory antibodies would restrict local searches from occurring around low scoring antibodies.

```
inputs: $n$          the number of iterations to perform
        $s$          size of main population
        $nFittest$   number of highest affinity Individuals to select for clonal
                     selection. Also, in metadynamics, indicates the number of
                     lowest affinity Individuals replaced by highest affinity clones
        $nWorst$     number of lowest affinity Individuals to replace by randomly
                     generated Individuals
        $cloneRate$  a multiplication factor of an Individual's affinity defining the
                     number of clones to produce during clonal selection. A
                     minimum of 1 clone is always produced.
begin
    $i \leftarrow 0$
    $P_i \leftarrow$ initPop($s$)                          // initialise main population
    $M \leftarrow \{\}$                                    // reset memory set, $M$
    while $i < n$ do
        calculateAffinity($P_i$)
        $L \leftarrow$ addToMemory($P_i$, $M$)     // $L$ = individuals added to memory
        $B \leftarrow$ selectFittest($P_i$, $nFittest$)      // select best Abs from $P_i$
        $R \leftarrow$ randomSelection($M$, $nFittest$)         // randomly select Abs
                                                               // from $M$
        $B \leftarrow$ combine($B$, $R$)                              // combine and
        $B_1 \leftarrow$ randomSelection($B$, $nFittest$)       // randomly select Abs
        $L \leftarrow$ combine($L$, $B_1$)
        $C \leftarrow$ clonalSelection($L$, $cloneRate$)
        addToMemory($C$, $M$)                      // add useful clones to memory
        $P_{i+1} \leftarrow$ metadynamics($P_i$, $C$, $nFittest$, $nWorst$)
    end
    return $M$                                                // return memory set
end
```

Algorithm 2. Immune Inspired Algorithm for Mutation Testing

In mutation testing, it is unlikely that a single test will kill all mutants. As such, the best GA/MT individual needs to possess enough tests to kill all mutants. *But how many is enough?*[3] The IIA/MT, on the other hand, naturally evolves a dynamic number of specialist memory individuals, each killing at least one mutant not killed by anything else. There is no need to predefine how many tests will be needed.

The evaluation mechanism of each algorithm's individual is effectively the same between both algorithms - the number of mutants an individual kills. Further differences however, occur in the adaptation mechanisms of the algorithms themselves. Whilst there is an argument that our immune algorithm is effectively a GA without crossover, there is an important difference between the purposes of their respective selection and mutation methods. GAs *select*

[3] This is discussed more in [4].

high fitness individuals because they are good solutions to the problem domain. Disregarding crossover, they randomly *mutate* child individuals to add diversity to the population - to search for new solutions or to escape local maxima. Our Immune algorithm on the other hand, evolves a set of specialist solutions. Their form of *selection*, as part of cloning, produces local populations around potentially good solutions. These clones are *mutated* to search around these, already good, solutions in an attempt to find better (higher affinity) solutions - local searches. Diversity - breadth searching - is added in a later stage, called metadynamics, by the death of individuals and the introduction of new, randomly generated individuals. Consequently, for an IIA, cloning and mutation are proportional to affinity (high affinity individuals undergo higher cloning rates and less mutation than low affinity ones), whereas for a GA, although selection is proportional to fitness, mutation is usually at a fixed rate (typically $< 10\%$).

5 Algorithm Comparison

Four programs have been tested (detailed in [4]): *DateRange* (**DR** - McCabe complexity: 6); *TriangleSort* (**TRI** - McCabe complexity: 11); *CalDay* (**CD** - McCabe complexity: 4); and, *Select* (**SEL** - McCabe complexity: 19). Each program was executed at least 30 times per algorithm. The initial population of either algorithm contains 300 tests (300 1-test individuals for the IIA/MT, 15 20-test individuals for the GA/MT), each initiated to the same value. Test values were randomly chosen that produced a low starting mutation score, and these same values (for a given program) were used for each experiment. This method was chosen over the random generation of each test as it allows the effects of each algorithm to be easily observed, with each algorithm starting from the same state - no bias is given to either algorithm from starting with a different population of tests. In total, 500 iterations of each algorithm were performed. Table 1 details the (non-optimised) parameter values used, representing typical values that may be chosen:

Table 1. Parameter values used for each algorithm

IIA/MT		**GA/MT**	
Parameter	Value	Parameter	Value
s (initial size of population)	300	s (size of population)	15
nFittest	5	m (#tests per member)	20
nWorst	5	crossRate	0.8
cloneRate	10	mutRate	0.02

The most relevant measure of algorithm effectiveness is determined by the time taken to generate a test set and the mutation score achieved; a more effective algorithm will achieve the same mutation score in less time. In this case however, time is more accurately measured across various algorithms by

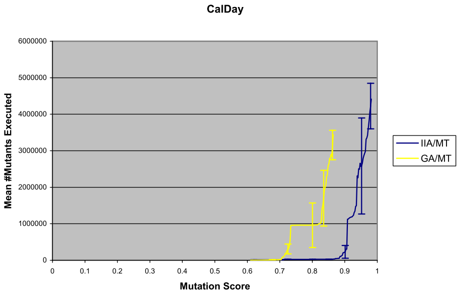


Fig. 3. Mean number of mutant executions to achieve at least a specific mutation score for the *CalDay* program

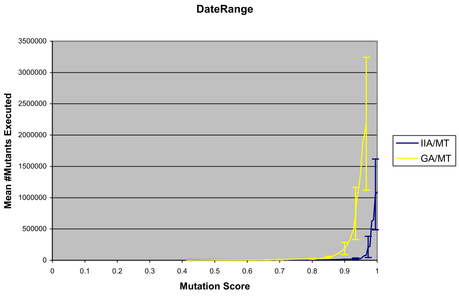


Fig. 4. Mean number of mutant executions to achieve at least a specific mutation score for the *DateRange* program

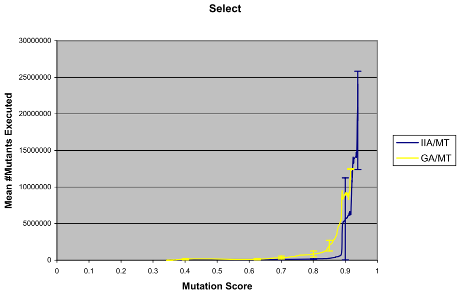


Fig. 5. Mean number of mutant executions to achieve at least a specific mutation score for the *Select* program

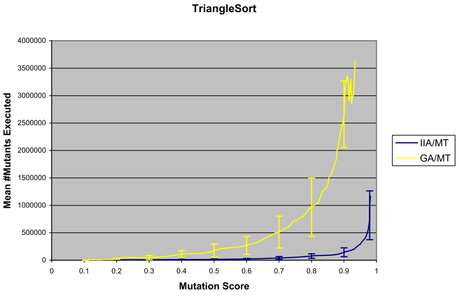


Fig. 6. Mean number of mutant executions to achieve at least a specific mutation score for the *TriangleSort* program

counting the number of mutant program executions, rather than iterations. Each algorithm performs differing operations per iteration, and so a single iteration of each algorithm will not necessarily achieve the same amount of work. However, each algorithm will need to execute the mutants against a number of tests. As long as mutant execution is done consistently across all algorithms, the number of executions to achieve a specific mutation score (regardless of how many iterations) forms a good measure of each algorithms effectiveness.

Figures 3-6 (*Error bars are* ± *1 s.d.*) show the mean number of mutants executed to achieve at least a specific mutation score for each of the four programs (some graphs display drops in mutation score; this is a result of a decrease in the number of runs achieving that mutation score affecting the overall mean). The results are similar for all four programs. During the initial stages the number of mutants executed remains low for both algorithms. As the mutation score increases, the number of mutants needing to be executed to improve the mutation score increases dramatically for the GA/MT before increasing for the IIA/MT. This suggests that in early iterations for both algorithms, weak mutants are killed by the introduction of new, weak tests - i.e. any test is capable of killing a mutant, resulting in a large increase in mutation score with few mutants executed. As the iterations progress however, the weaker mutants are removed leaving only the stronger ones. The tests needed to kill these are harder to generate meaning more mutants are executed before a test is found to improve the mutation score. The graphs show that an IIA/MT is able to generate more of these “harder” tests in fewer executions, leading to higher mutation scores for the same number of executions. Alternatively, an IIA/MT needs fewer executions to achieve the same mutation score.

The GA/MT execution of *CalDay* (Figure 3) is distinctive from the other programs in that at a mutation score of approximately 73%, the mutation score suddenly increases rapidly to approximately 82% in relatively few mutant executions, before continuing as is “standard”. This behaviour is different from that exhibited by the other programs, and although the results coincide with a decrease in the number of runs achieving these (and higher) mutation scores (1 run less), it is unlikely this decrease caused the effect seen - if it was the reason, the effect would more likely have been an increase in mutation score, with an oscillation in the number of executions, as seen in figures 6 and 5. A reasonable explanation would be that specific tests are needed to kill 73% of mutants - possibly they enter a specific portion of code that had not been reached before. These tests may also kill further mutants, improving upon the mutation score with no extra executions. Upon further inspection, this effect is not apparent for the IIA/MT, however the higher granularity graph does show the same effect, supporting the notion that this is a result of this program's test data development, and not some curiosity from the GA/MT algorithm itself [4].

Table 2. T-test (0.05 level) results for the significance between the mean number of mutants executed for each algorithm, at the highest mutation score obtained by both algorithms in at least 25 runs. Results are rounded up to the nearest integer to reflect that a mutant is either executed completely or not at all.

	MS	IIA/MT mean	GA/MT mean	t_{obt}	t_{crit}
DR	94.44%	24372 ± 12991	1188082 ± 706193	9.02	2.00
TRI	88.95%	125802 ± 74347	2339044 ± 741550	17.34	2.00
CD	84.63%	27551 ± 4953	2341483 ± 709736	18.78	2.00
SEL	88.97%	1733911 ± 1664843	8920678 ± 2767622	12.16	2.00

T-tests at the 0.05 level, to check the significance of the difference in mean number of mutants executed, are shown in table 2. They reject the null hypothesis for all four programs. These tests were performed at the equally highest mutation score achieved by at least 25 runs[4] of both algorithms.

Ultimately then, it would appear that the IIA/MT's method of generating tests outperforms (in terms of number of executions) the GA/MT's. Foremost, the GA/MT has a finite number of tests, per individual, with which to generate a

[4] Comparisons are made where at least 25 runs have achieved the result so that reliable mean averages (and T-test results) can be calculated. Obviously a higher mutation score may be possible using an algorithm, just fewer than 25 runs achieve it.

full mutation score. If every one of the 20 tests kills at least one distinct mutant, then all 20 tests are required. Now, if the individual's mutation score is less than 100% then a further test is required. Unfortunately there is no room to add it into the individual, and so one of the existing tests must be improved - something which may not be possible. The IIA/MT on the other hand, has no memory set size restriction, and is free to add additional tests that only kill a single mutant. This advantage is complemented by the IIA/MT's local and global search facilities: local searching occurs through cloning and mutating tests proportional to their mutation scores, which could be considered as similar to searching around input boundaries; global searching happens by randomly introducing a number of tests each iteration.

6 Conclusions

This paper investigates the hypothesis that an Immune Inspired Algorithm for Mutation Testing (IIA/MT) is consistently at least as effective at evolving test data as a Genetic Algorithm for Mutation Testing (GA/MT). T-test results in table 2 reject the null hypothesis for the four programs tested, within 500 iterations, providing evidence to suggest a significant difference in the mean number of mutants executed to achieve at least a specific mutation score. Evidence suggests that using the IIA/MT is favourable to the GA/MT, particularly for higher mutation scores. In particular, a drawback of the GA/MT is that it has fixed sized individuals which can limit the achievable mutation score. The IIA/MT does not suffer from this problem.

References

1. Beizer, B.: Software Testing Techniques, 2nd edn. VN Reinhold, New York (1990)
2. Wong, W.E.: On Mutation and Data Flow. PhD thesis, Purdue University (1993)
3. May, P., Mander, K., Timmis, J.: Software vaccination: An artificial immune system approach. In: Timmis, J., Bentley, P.J., Hart, E. (eds.) ICARIS 2003. LNCS, vol. 2787, pp. 81–92. Springer, Heidelberg (2003)
4. May, P.: Test Data Generation: Two Evolutionary Approaches to Mutation Testing. PhD thesis, The University of Kent at Canterbury (2007)
5. Offutt, A.J., Untch, R.H.: Uniting the Orthogonal. Kluwer, Dordrecht (2000)
6. Baudry, B., Fleurey, F., Jezequel, J.M., Traon, Y.L.: Genes and bacteria for automatic test cases optimization in the.net environment. In: Proc. of Int. Symp. on Software Reliability Engineering (ISSRE 02), pp. 195–206 (2002)
7. Mitchell, M.: An Introduction to Genetic Algorithms, 6th edn. MIT Press, Cambridge (1999)
8. de Castro, L.N., Zuben, F.J.V.: The clonal selection algorithm with engineering applications. In: Proc. GECCO, pp. 36–37 (2000)

An Artificial Immune System Based Approach for English Grammar Checking

Akshat Kumar and Shivashankar B. Nair

Indian Institute of Technology Guwahati, India
akshat.kumar@gmail.com, sbnair@iitg.ernet.in

Abstract. Grammar checking and correction comprise of the primary problems in the area of Natural Language Processing (NLP). Traditional approaches fall into two major categories: Rule based and Corpus based. While the former relies heavily on grammar rules the latter approach is statistical in nature. We provide a novel corpus based approach for grammar checking that uses the principles of an Artificial Immune System (AIS). We treat grammatical error as pathogens (in immunological terms) and build antibody detectors capable of detecting grammatical errors while allowing correct constructs to filter through. Our results show that it is possible to detect a range of grammatical errors. This method can prove extremely useful in applications like Intelligent Tutoring Systems (ITS) and general purpose grammar checkers.

1 Introduction

Grammar checking and correction constitute two of the major steps in Natural Language Processing (NLP). Their importance ranges from base applications, like using a word processor, to highly specialized tasks like transforming high level natural language commands into machine understandable forms. There are two well-known approaches for grammar checking namely the Rule-based ones and the pattern-matching-corpus-based approaches. Existing grammar checking systems, such as those described in [10], [1], [6], [4], fall into the former category, addressing the issue with a collection of heuristic rules that approximate a natural language grammar. The other approach is based on the application of corpus linguistics to the task of language processing [11].

While the rule based approach focuses on understanding the grammar of natural language, the corpus based approaches try to statistically analyze the language by taking the advantage of the abundance of available text. Our approach falls in this second category and is unique in the respect that it uses an Artificial Immune System (AIS) based technique. The motivation for our approach comes from the human immune system which is able to distinguish every harmful external entity from the self cells of the human body. We have modeled our grammar checker based on a similar approach so that it is able to identify any entity outside the corpus (regarded as error). The self in our case is the corpus itself.

In section 2 we describe briefly the basics of an Artificial Immune System. Section 3 focusses on how we adapt the AIS for the task of natural language processing. Section 4 describes how we generate antibodies for the grammar checker. Section 5 evaluates our approach on a collection of grammatical errors while Section 6 raises light on the

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 348–357, 2007.

limitations of our approach. Finally section 7 portrays the conclusions arrived at and future enhancements that could be made.

2 Artificial Immune System (AIS)

While the biological immune system generates antibodies to detect and defend the body of the being, its counterpart the artificial immune system (AIS) works on principles and algorithms laid down from theoretical immunology to evolve solutions for a range of problems. A good description of the biological and artificial sides of the immune system can be found in [3] and [9]. We describe the working of an AIS in brief.

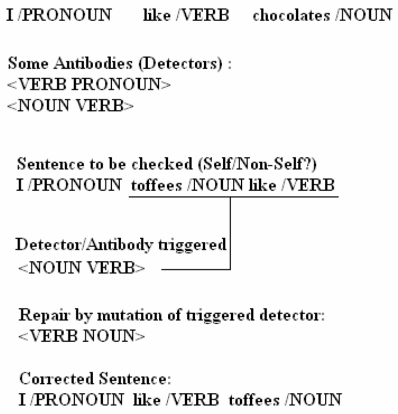
Our human immune system is based on a collection of immune cells called *lymphocytes*. These primarily constitute the B-cells and the T-cells. Both these cell types present receptor molecules on their surfaces responsible for recognizing the antigenic patterns displayed by pathogens. The main role of a lymphocyte in an immune system is encoding and storing a point in the solution space or shape space. The match between a receptor and an antigen may not be exact and it takes place with a strength termed as affinity. If this affinity is high the antigen is said to be within the lymphocyte's recognition region.

After the successful recognition of the harmful pathogens an adaptive immune response is invoked. In this response those cells that were capable of identifying the pathogens (non self) proliferate by cloning. They may also undergo controlled mutation (hypermutation)[3] so as to fine tune their receptor molecules resulting in an increase in affinities. A selective mechanism guarantees that those offspring cells (in the clone) that better recognize the antigen and which elicited the response have longer life spans. These cells are called Memory cells. These memory cells are the ones that quickly identify the disease causing organism in future attacks and thus trigger a faster secondary immune response. The whole process of antigen recognition, cell proliferation and differentiation into memory cells is called clonal selection.

3 Adapting AIS for Language Processing

The idea of adapting an AIS to the field of NLP comes from the commonly observed fact that a person conversant in a language generally finds it difficult to generate an incorrect sentence in that language. It is argued here that a person fluent in a natural language has already built an immune system to detect and reject incorrect sentences in that language. In short he is immune to an incorrect language generation attack which is why he experiences difficulty in generating examples of incorrect sentences at the same frequency as correct ones. Grammar by itself is never talked off in the initial phases of language learning. A child picks up a language oblivious of grammar. A collection of sentences, that constitute a corpus, is formed initially and subsequent sentence generation largely depends on the combinations of words and grammar contained within it. The corpus thus could be viewed as the collection of correct sentences constituting a collection of the self cells (cases). Detectors could now be generated by constructs that do not exist within this corpus. Any form of grammar not found in the corpus could be treated as an error and hence could be a candidate to mature into a detector. The underlying assumption that the corpus is complete, however has to hold.

Based on this we propose an architecture for detecting and modifying grammatical errors based on applying the anomaly detection capabilities of an AIS. In our approach we treat the grammatical errors in the sentences as metaphors for the non-self antigens which in turn elicit a response from the immune system viz. the grammar checker described herein. To catch the harmful antigens (grammatical errors) in the system a repertoire of antibodies is to be generated. This generation is based on the method described in [2] with modifications made to suit the language processing scenario. Figure 1 shows how the antibody and the antigen interact in the domain of languages. In the following sections, we describe the definitions of self and non-self followed by the antibody generation technique for grammar checking.



I /PRONOUN like /VERB chocolates /NOUN

Some Antibodies (Detectors) :
<VERB PRONOUN>
<NOUN VERB>

Sentence to be checked (Self/Non-Self?)
I /PRONOUN toffees /NOUN like /VERB

Detector/Antibody triggered
<NOUN VERB>

Repair by mutation of triggered detector:
<VERB NOUN>

Corrected Sentence:
I /PRONOUN like /VERB toffees /NOUN

Fig. 1. Simple case of formation of antibodies, detection and correction of a sentence

3.1 Self

The set of self in our system is an extensively part of speech tagged corpus. Corpus is analogical to the human body and whatever constitutes the corpus is similar to the body cells or self cells. It may be possible that a valid grammatical structure be flagged as an error if it is outside the corpus. Given a sufficiently large corpus our system can perform satisfactorily covering a large number of grammatical constructs.

The Corpus. We are using the corpus Reuters-21578, Distribution 1.0 text collection. This corpus is a collection of various news documents appeared on the Reuters newswire in 1987. We have built a parser that can parse this corpus and extract meaningful text from it. The parser filters out documents in cryptic format like share market information and other economics related information to produce this meaningful text.

Part of Speech Tagging. The Reuters-21578 corpus is tagged for part of speech by using MontyLingua parser available at MIT [7]. This parser extracts extracts subject/verb/

object tuples, extracts adjectives, noun phrases and verb phrases, and extracts people's names, places, events, dates and times, and also does part of speech tagging from the English sentences within the corpus.

Self Structure. The Self is constituted by the bigrams, trigrams and tetragrams from the POS tagged corpus. The following example makes this process clearer.

Sentence 1. *Officials/NNS were/VBD not/RB immediately/RB available/JJ for/IN comment/NN ./.*

- *Bigrams: <NNS,VBD>, <VBD,RB>...*
- *Trigrams: <NNS,VBD,RB>, <VBD,RB,RB>...*
- *Tetragram: <NNS,VBD,RB,RB>, <VBD,RB,RB,JJ>...*

The tag-set used has only 36 members and this makes the system very general to properly model the underlying grammar of language. Therefore to capture more information about the actual language we have extended the tag-set by incorporating the use of some of the regular English words as the tags. By studying the existing tag-set, we have noticed that a single tag covers many English words. For example the tag DT covers articles such as *a, an, the, these, those* etc. To make fine distinction in the grammar usage we treat all these words as different tags. A list of words which should be treated differently was made based on the studies carried out on the Penn Treebank tag-set.

Error Detection. The basic units of the sentence which are analyzed are the bigram, the trigram and the tetragram of the part of speech sequences and of extended tags sequences in the test sentence. These components were tested against the Antibodies of our AIS. These Antibodies were generated to capture only the non-self (ungrammatical) entities. Once the antibodies recognized an antigen (or grammatical error) the system flags off the corresponding word sequence as one containing an error.

In the following section we describe the antibody generation phase of our AIS.

4 Generating Antibodies for Grammar Checking

This section describes the antibody generation phase. We use the Negative Selection algorithm [5] which is based on the principles of self-nonself discrimination in the biological immune system (see figure 2). This negative selection algorithm can be summarized as follows:

- Define *self* as a collection **S** of elements in a feature space **U**, a collection that needs to be monitored. For instance, if **U** corresponds to the space of states of a system represented by a list of features, **S** can represent the subset of states that are considered as normal for the system.
- Generate a set **F** of detectors, each of which fails to match any string in **S**. An approach that mimics the immune system generates random detectors and discards those that match any element in the self set.
- To monitor the new data, continuously check it against the generated set of detectors and if any detector matches regard it as an anomaly or error.

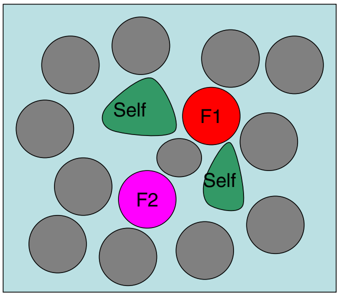


Fig. 2. The figure illustrates the concept of self and non-self in a feature space. F1 and F2 etc. indicate different fault condition represented by detectors.

4.1 Real Valued Negative Selection Algorithm (RNS)

The RNS detector generation starts with a population of candidate detectors, which are then matured through an iterative process. In particular, the center of each detector is chosen at random and the radius is a variable parameter which determines the size (in the m-dimensional space) of the detector. The basic algorithmic steps of the RNS detector generation algorithm are given in Figure 3.

At each iteration, the radius of each candidate detector is calculated, and the ones that fall inside *self* region are moved (i.e. its centre is successively adjusted by moving it away from training data and existing detectors). The set of non-self detectors are then stored and ranked according to their size (radius). The detectors with larger radii (and smaller overlap with other detectors) are considered as better-fit and selected to go over the next generation. Detectors with very small radii, however, are replaced by the clones of better-fit detectors. The clones of a selected detector are moved at a fixed distance in order to produce new detectors in its close proximity. New areas of the non-self space are explored by introducing some random detectors. The whole detector generation process terminates when a set of mature (minimum overlapping) detectors are evolved which can provide significant coverage of the non-self space.

A **detector** is defined as $d = (c, r_d)$, where $c = (c_1, c_2, .., c_m)$ is an m-dimensional point that corresponds to the center of a hyper-sphere with r_d as its radius. In our system we use numeric references for tags, the self-set for our system consists of 2D (for bigram sequences) points, 3D (for trigram) points and 4D (tetragram) points. The following parameters are used for the detector generation process:

- r_s The threshold value (allowable variation) of a self point; in other words, a point at a distance greater than or equal to r_s from a self sample is considered to be abnormal. In our case since we want a strict checking for errors, this parameter would be zero.
- α The variable parameter to specify the movement of a detector away from a self sample or existing detectors.
- γ The maximum allowable overlap among the detectors, which implicates that allowing some overlap among detectors can reduce holes in the non-self coverage.

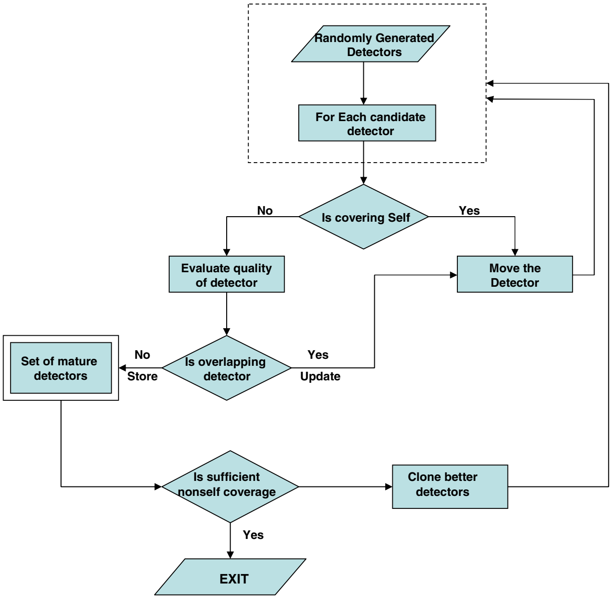


Fig. 3. Flow diagram showing the algorithmic steps for real valued negative selection algorithm

Calculating the Detector Radius. We have used the Euclidean distance to measure the distance between two points x and y, which is defined as: $D(x, y) = (\Sigma|x_i - y_i|^\lambda)^{1/\lambda}$

Where $x = x_0, x_1, ..x_n$ and $y = y_0, y_1, ..y_n \; with \; \lambda = 2$

This approach allows having variable size detectors to cover the non-self space. As shown in Figure 4(a), if the distance between a candidate detector, $d = (c, r_d)$ and its nearest self point in the training dataset is D, then the detector radius is considered as $r_d = (D - r_s)$.

Moving the Detector. Let $d = (c, r_d)$ represents a candidate detector and $d^{nearest} = (c^{nearest}, r_d^{nearest})$ is its nearest detector (or a self point), then the center of d is moved such that

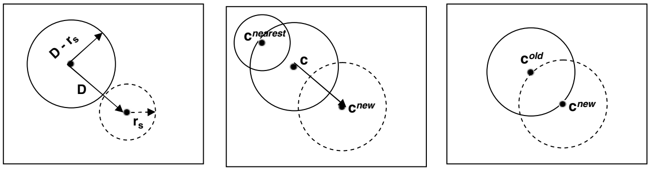
$$c^{new} = c + \alpha * dir/(||dir||) \tag{1}$$

where $dir = c - c^{nearest}$, and $||\ \ ||$ denotes the norm of an m-dimensional vector. Accordingly, if a detector overlaps significantly with any other existing detectors, then it is also moved away from its nearest neighboring detector.

Detector Cloning and Random Exploration. At every generation, a few better- fitted detectors are chosen to be cloned. Specifically, let $d = (c^{old}, r_d^{old})$ be a detector to be cloned and, say $d^{clon} = (c^{clon}, r_d^{clon})$, is a cloned detector whose centre is located at a distance r_d^{old} from d and whose radius is the same as that of the detector, d. Accordingly, the centre of d^{clon} is computed as

$$c^{clon} = c^{old} + r_d^{old} * dir/(||dir||) \tag{2}$$

Where $dir = c^{old} - c^{nearest}$ (where $c^{nearest}$ is the center of d's nearest detector)



(a) Calculation of the detector radius (b) Movement of the candidate detector (c) Cloning of a detector

Fig. 4. Steps in detector maturation process

Evaluation of Non-self Detectors. Detectors which do not fall in the *self* region are sorted according to their size. A detector with large radius gets selected for the next generation population, if it has small overlap with existing detectors i.e. less than the overlapping threshold. The overlapping measure W of a detector is computed as the sum of its overlap with all other detectors as follows

$$W(d) = \sum_{d \neq d'} w(d, d') \tag{3}$$

where $w(d, d')$ is the measured overlap between two detectors $d = (c, r_d)$ and $d' = (c, r_{d'})$; and is defined by

$$w(d, d') = (exp(\delta) - 1)^m, \text{ m is the dimension of the feature space} \tag{4}$$

$$and\ \delta = (r_d + r_{d'} - D)/2r_d \tag{5}$$

The value of δ is considered to be bounded between 0 and 1; and D is the distance between two detector centers c and c'. This overlapping measure seems to favor the detectors with bigger radii, i.e. detectors having larger coverage of the non-self space with minimum overlap among them.

5 Experimental Results

We have implemented the grammar checker as described in previous sections in JAVA and tested the same using test sentences from a book of grammatical errors [8]. The corpus for our checker is the Reuters-21578, Distribution 1.0 text collection available freely on the web. The list of these error types is provided below with sample sentences denoting errors.

We have found out eight categories of grammatical errors which our grammar checker can detect. However it is to be noted that it may not be able to correct all sentences of these categories as some of the finer distinction of a construct may be absent in the corpus. The incorrect constructs in the sentences are caught by the antibodies in our grammar checker while correct constructs are undetected.

5.1 Subject Verb Disagreement

Sentence 2. *These lines occurs in Shelleys poem. (DT NNS VBZ IN NN .)*

- ***Error*** *These lines occurs in*
- ***Correct*** *These lines occur in Shelleys poem.*

5.2 Choice of Tense

Sentence 3. *Those boys fight.(DT NNS VBP .)*

- ***Error*** *those boys fight*
- ***Correct*** *Those boys are fighting.*

Sentence 4. *I am hearing a noise outside. (PRP VBP VBG DT NN IN .)*

- ***Error*** *I am hearing a*
- ***Correct*** *I hear a noise outside.*

5.3 Article Misuse

Sentence 5. *Are you in hurry? (VBP PRP IN NN .)*

- ***Error*** *you in hurry*
- ***Correct*** *Are you in a hurry?*

5.4 Wrong Pronouns

Sentence 6. *Who of the two girls look good. (WP IN DT CD NNS VBP JJ .)*

- ***Error*** *Who of the two*
- ***Correct*** *Which of the two girls look good?*

5.5 Wrong Numbers

Sentence 7. *I lost my baggages in the train. (PRP VBD PRP NNS IN DT NN .)*

- ***Error*** *I lost my baggages*
- ***Correct*** *I lost my baggage in the train.*

5.6 Wrong Adverbs

Sentence 8. *We reached station timely. (PRP VBD NN JJ .)*

- ***Error*** *reached station timely*
- ***Correct*** *We reached station in time.*

5.7 Wrong Adjectives

Sentence 9. *This is most unique institution. (DT VBZ RBS JJ NN .)*

- ***Error*** *is most unique institution*
- ***Correct*** *This is unique institution.*

5.8 Missing Verb

Sentence 10. *I going home. (PRP VBG NN .)*

- ***Error*** *I going*
- ***Correct*** *I am going home.*

6 Limitations

Although our grammar checker catches a variety of constructs, experimentally we found out that many ungrammatical constructs are passed undetected. This deficiency can be traced to the limited tag-set being used in this work. Being small, this tag set is incapable of modeling the entire underlying English grammar. However this is not a shortcoming of our approach. If we use a more enhanced tag-set like the C7 tag-set used to annotate the British National Corpus then we can identify the different usage of words like *I* and *He* (treated same in current approach) because these two are assigned different tags PPIS1 and PPHS1 respectively.

The current system can not detect differences between singular and plural forms of a word (mostly pronoun). For example word *they* and *He* are given same tag PRP. Again this difficulty can be solved by using a more enhanced tagset.

In addition we still need to have some connection between the actual words used in the corpus and the part of speech tags. Currently our grammar checker only uses the N-gram sequences. To make a checker with high precision we would also have to incorporate these words in our error checking process.

7 Conclusion and Future Work

The motivation for our approach comes from the human immune system which is able to identify almost every external harmful entity that encroaches the human body. We have modeled our grammar checker on such an approach so that it is able to identify any grammatical construct outside the corpus and flag it as an error. The corpus constitutes the self of our system. Although this grammar checker is not able to trap all the errors, it may be inferred that the algorithms innately used by the human immune system are definitely effective as one level of detection.

Being language independent is one major advantage of our approach. Grammar rules for the language need not be provided *a priori*. Thus during the error detection phase no grammar rules are used to generate the antibodies or error detectors. The generation of antibodies is dependent only on the *self* set.

A major improvement can be the addition of the ability to record the antibodies or error detector usage information. This information would help us to identify those antibodies which are used repeatedly and in a way indicative of the most common mistakes committed by a person. This knowledge about a language learner's mistakes can provide valuable tips in many scenarios of pedagogy. Consequently our system can also be used as a plug-in for an intelligent language tutoring system for Non-Native speakers of English or for that matter other languages. Once a user interacts with the system for a sufficient amount of time the detector usage information can easily bring out the weaknesses of that person.

References

1. Bolioli, A., Dini, I., Mahmti, G.: Jdii: Parsing italian with a robust constraint grammar. In: Proceedings of the 15th International Conference on Computational Linguistics, pp. 1003–1007 (1992)
2. 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. 21–35. Springer, Heidelberg (2004)
3. de Castro, L.N., Timmis, J.: Artificial immune systems: A new computational intelligence approach. Springer, London (1992)
4. Flora, B., Fernando, R.: Gramcheck: A grammar and style checker. In: Proceedings of the International Conference on Computational Linguistics (COLING), pp. 365–370 (1996)
5. Forrest, S., Perelson, A., Allen, L., Cherukuri, R.: Selfnonself discrimination in a computer. In: Proceedings of the IEEE Symposium on Security and Privacy, IEEE Computer Society Press, Los Alamitos (1994)
6. Genthial, D., Courtin, J.: From detection/correction to computer aided writing. In: Proceedings of the International Conference on Computational Linguistics (COLING), pp. 1013–1018 (1992)
7. Liu, H.: Montylingua: An end-to-end natural language processor with common sense (2004)
8. Misra, A.K.: Avoid errors (1994)
9. Sompayrac, L.: How the immune system works. Blackwell Science, Oxford (1999)
10. Thurmair, G.: Parsing for grammar and style checking. In: Proceedings of the International Conference on Computational Linguistics (COLING), pp. 365–370 (1990)
11. Tschichold, C., et al.: Developing a new grammar checker for english as a second language. In: Technical Report Laboratoire de traitement du langage et de la parole

A Novel Clonal Selection Algorithm Based Fragile Watermarking Method

Veysel Aslantas[1], Saban Ozer[2], and Serkan Ozturk[1]

[1] Erciyes University, Engineering Faculty, Computer Engineering Div.,
38039 Kayseri Turkey
[2] Erciyes University, Engineering Faculty, Electrical-Electronics Engineering Div.,
38039 Kayseri Turkey
{aslantas,sozer,ozturks}@erciyes.edu.tr

Abstract. In this paper, a novel fragile watermarking method based on clonal selection algorithm (CSA), CLONALG, is presented. In Discrete Cosine Transform (DCT) based fragile watermarking techniques, there occurs some degree of rounding errors because of the conversion of real numbers into integers in the process of transformation of image from frequency domain to spatial domain. In this paper, the rounding errors caused by this transformation process are corrected by using CLONALG. Simulation results show that extracted watermark is obtained exactly the same as embedded watermark and optimum watermarked image transparency is achieved. In addition, the performance comparison of CLONALG and genetic algorithm (GA) based methods is realized.

Keywords: Fragile image watermarking, discrete cosine transform, clonal selection algorithm, multimedia.

1 Introduction

Due to advances in computer and communication networks, digital media including audio, image and video can be easily transmitted via the Internet. Transmitted data can also be easily altered, copied or even stolen. Therefore, digital media copyright protection has become a great challenge. Digital watermarking is one of the methods to provide security in copyright protection.

Digital watermarking is to embed secret information, or the watermark, into digital media data. Content providers want to embed watermarks in their digital media data (multimedia objects, digital content) for several reasons like content authentication, fingerprinting, meta-data insertion, copyright protection, content archiving, broadcast monitoring and tamper detection [1].

Digital image watermarks can be classified as “visible” and “invisible”. Visible watermarks are visual patterns like the logos embedded into one corner of an image [2]. Although the logos or watermarks are easily identified, they can be removed or destroyed easily. The invisible watermarks are hidden on the unknown places of the image and are more robust than the visible watermarks. Only the authorized persons can extract the embedded watermark.

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 358–369, 2007.

Furthermore, digital watermarks can also be categorized as “robust”, “semi-fragile” and “fragile” [3]. Robust watermarks are designed to resist some image processing operations (image scaling, cropping, lossy compression, etc.), called attacks. Semi-fragile watermarks are capable of tolerating some degree of change of a watermarked image, such as the addition of quantization noise from lossy compression [4]. Fragile watermarks are designed to detect unauthorized modifications on the watermarked images. Therefore, fragile watermarking methods are mainly used for the purpose of image authentication in the area of satellite or medical imagery.

Digital watermarking techniques can be further classified into two categories: spatial domain and transform domain. In the spatial domain, watermarks can be embedded by modifying the pixel values. In the transform domain, they can be embedded by modifying the transformation coefficients of DCT, Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT), etc.

In recent years, digital watermarking techniques have been improved using optimization algorithms such as GA which is a popular evolutionary optimization technique invented by Holland [5]. This algorithm can find the global optimal solution in complex multidimensional search spaces. GA is modelled on natural evolution in that the operators it employs are inspired by the natural evolution process. These operators, known as genetic operators, manipulate individuals in a population over several generations to improve their fitness gradually [6]. Kumsawat et al.[7] proposed a watermarking technique for optimizing the image watermarking using the GA. They applied GA to wavelet transform domain to improve the quality of the watermarked image and the robustness of the watermark. Kumsawat [8] also proposed the spread spectrum image watermarking algorithm based on the discrete multi-wavelet and GA. Similar to their previous technique, parameters consisting of threshold values and the embedding strength are searched using GA to improve the quality of the watermarked image and the robustness of the watermark. Wang et al. [9] presented VQ-based watermarking technique for hiding the gray watermark. They applied the VQ coding procedure with GA to compress the original watermark into smaller size and embedded the coded results into the cover image in the VQ domain. A new watermarking technique based on the DCT and the GA is proposed by Shieh et al. [10]. GA is used to search for the optimum frequency bands in the DCT coefficient blocks to embed the watermark for the improvement of security, robustness and the quality of the watermarked image. Chang et al. [11] proposed a new DCT based watermarking method combining ART2 neural network with GA. ART2 neural network was used to classify 8×8 DCT blocks of images in training sets. The optimal coefficients for watermark embedding were found by using GA for each cluster. Shih et al. [12] proposed a novel fragile watermarking technique based on DCT and GA. In this paper, the rounding errors caused by transformation of frequency domain to spatial domain are reduced by using GA. Aslantas [13] proposed a novel optimal watermarking scheme based on singular-value decomposition using GA.

In DCT based fragile watermarking techniques, the original input image is first transformed into its frequency domain. Then, the watermarks are generally embedded by modifying the least significant bits (LSBs) of the frequency domain coefficients. After embedding process is completed, there occurs some degree of rounding errors because of the conversion of real numbers into integers in the process of transformation of image from frequency domain to spatial domain. In this paper, a new watermarking method based on CSA CLONALG is developed to reduce these rounding errors. CSA was first proposed by de Castro and Von Zuben [14] and was later enhanced and named as CLONALG [15].

This paper is organized as follows: In Section 2, the fundamental concepts of CSA is described. Section 3 demonstrates the CSA based method which reduces the rounding errors. In Section 4, simulation results and comparison of the algorithms are demonstrated. Finally, Section 5 is the conclusion part of this paper.

2 Fundamental Concepts of CSA

The clonal selection principle is used to explain the basic features of an adaptive immune response to an antigenic stimulus [16]. In the adaptive immune system, any molecule that can be recognized is called as an Antigen (*Ag*). Lymphocytes are the basic immunity cells. They are called "*T*" and "*B*" cells. Unlike the "T" cells which need the recognition of antigens via other assisting cells, the "*B*" cells can recognize the antigens without restraint in liquid solutions. Antibodies (*Ab*'s) are molecules attached to the surface of *B* cells the aim of which is to recognize and bind to *Ag*'s. When binding to these *Ab*'s with a second signal from accessory cells, such as the T-helper cell, B cell is stimulated by *Ag* to proliferate and mature into non-dividing *Ab* secreting cells, known as plasma cells. This process of cell division generates a clone known as a cell or set of cells that are the progenies of a single cell.

While proliferating and differentiating into plasma cells, the *B* cells can also differentiate into long-lived *B* memory cells. When memory cells are exposed to a second antigenic stimulus, commence to differentiate into plasma cells capable of producing high-affinity *Ab*'s, pre-selected for the specific *Ag* that had stimulated the primary response, they circulate through the blood, lymph, and tissues. Fig. 1 shows the clonal selection principle [17].

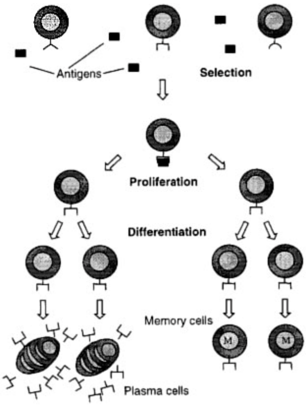


Fig. 1. Clonal selection principle

The main features of the clonal selection theory are [17]:

1) proliferation and differentiation on stimulation of cells with *Ag*'s;
2) generation of new random genetic changes, expressed subsequently as diverse *Ab* patterns, by a form of accelerated somatic mutation (a process called affinity maturation);
3) estimation of newly differentiated lymphocytes carrying low-affinity antigenic receptors.

Although it is not an exact copy of the immune system's behaviour, the computational model of the clonal selection principle borrows heavily (largely) from immunological theory as well as many other artificial immune system models.

For the maturation of the immune response, a base population of *Ab*'s is immunized with a training set of *Ag*'s. In the initial exposure, a set of memory cells can respond more quickly to the second construction of the *Ag*'s. By this way, the cells with the *Ag*'s can recognize both specific immunized cells and structurally similar ones, which is known as cross-reactive [18].

The steps of the clonal selection algorithm, CLONALG, are given below:

1) Generate a random initial population of antibodies, *Ab*'s.
2) Create a set of antigens *Ag*'s.
3) Select an antigen Ag_i from the population *Ag*.
4) Calculate the affinity values of every *Ab* population to Ag_i.
5) Select the *n* highest affinity antibodies and generate a number of clones (C^i) for them.
6) Mutate the clone population C^i to produce a mature population C^{i*}.
7) Apply the affinity function to each member of the population C^{i*} to select the best *n* members which become the new memory cell and remove low affinity ones.
8) Repeat steps 3-7 until all antigens have been presented. This represents one generation of the algorithm.

The algorithm given above can be summarized as follows: Firstly, an initial population of *Ab*'s is generated randomly in a given bounds for the problem (*Ag*) considered. *Ab* is represented by binary bit arrays. The length of the bit arrays is determined by the desired sensitivity. Then, the fitness value of each population member is calculated and the *Ab*'s are ordered according to their fitness values. *n* highest affinity *Ab*'s are selected and clonned according to fitness value proportionately. The number of clones generated for all these selected *Ab*'s is given by;

$$N_c = \sum_{i=1}^{n} round\left(\frac{\beta N}{i}\right) \tag{1}$$

where N_c is the total number of clones generated for each of the *Ag*'s, β is a multiplying factor, N is the total number of *Ab*'s, and $round(.)$ is the operator that rounds its argument toward the closest integer. Constructed clones are mutated inversely proportional with the fitness values. The fitness values of the mutated (mature) clones are calculated. Each *Ab* and its clones construct a sub-population. The highest fitness valued *Ab* of each sub-population is selected and permitted to live. *d* low affinity *Ab*'s of population are altered with the random generated *Ab*'s. Thus, the differentiation of the population is obtained. These steps continue until stopping criteria is reached.

3 CSA Based Watermarking Method

The watermark embedding and extracting methods used in this paper are similar to [12] and are described below.

Let the original input image be *I* with size $M \times N$ and the binary watermark image be W with size $M_w \times N_w$. In order to embed a watermark, the spatial domain pixels are needed to be transformed into DCT domain frequency bands. 8×8 block DCT is performed on *I* and the coefficients in the frequency bands (I^{DCT}) are obtained as given below.

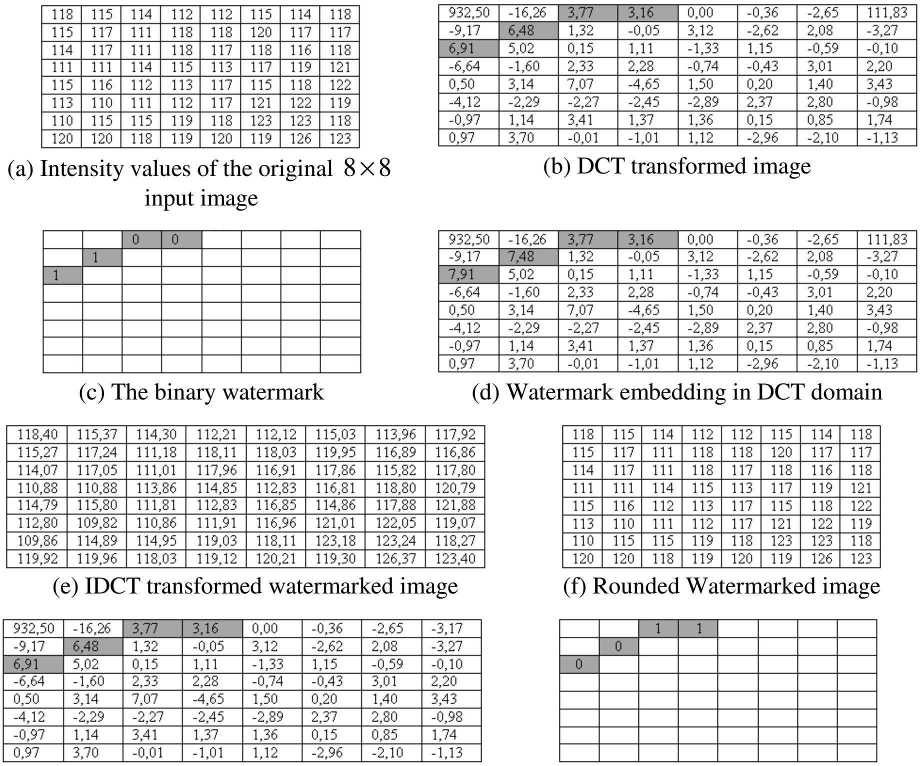
$$I^{DCT} = DCT(I) , \qquad I^{DCT} = \bigcup_{m=1}^{M/8} \bigcup_{n=1}^{N/8} I^{DCT}_{(m,n)} \tag{2}$$

For each 8×8 block in I^{DCT}, only four coefficients in frequency bands are modified for watermark embedding. The watermark embedding positions are $I^{DCT}_{(m,n)}(1,3)$, $I^{DCT}_{(m,n)}(2,2)$, $I^{DCT}_{(m,n)}(3,1)$ and $I^{DCT}_{(m,n)}(4,1)$, respectively. The embedding method is based on the Least Significant Bit (LSB) modification. The watermark is embedded into the integer part of absolute real number of I^{DCT}, and I^{WDCT} is obtained. Next, inverse DCT (IDCT) operation is applied to I^{WDCT} to obtain I^{WR}, which is the watermarked real number image. Then, all real numbers in I^{WR} are translated into integers and watermarked image (I^{W}) is obtained.

Watermark extracting method is also based on LSB modification as watermark embedding method. In order to extract the watermark, first, watermarked image is transformed into DCT domain. Then, the integer parts of absolute values belonging to specific positions (where the watermark is embedded) of the DCT domain are obtained. Then, these decimal values are translated into binary format. And finally, LSBs of the obtained binary values reveal the watermark.

In the watermarking method described above, there occurs some degree of rounding errors because of the conversion from real numbers to integers in the process of transformation of image from frequency domain to spatial domain.

An example of watermark embedding and extracting operation is shown in Fig. 2. The figure also illustrates the errors caused by using simple rounding technique in translating real numbers to integers. Fig. 2 (a) is original 8×8 gray-level input image in the spatial domain. Fig. 2 (b) is the DCT transformed image of Fig. 2 (a). Fig. 2 (c) is a binary watermark. Fig. 2 (d) is obtained by embedding Fig. 2 (c) into Fig. 2 (b) based on LSB modification. Fig. 2 (d) is transformed into its spatial domain by IDCT, and by this way Fig. 2 (e) is obtained. All pixels of Fig. 2 (e) are real numbers. These real numbers are rounded to integers as shown in Fig. 2 (f) which represents the watermarked image. In order to extract the watermark, Fig. 2 (f) is transformed into DCT domain and Fig. 2 (g) is obtained. Finally, the watermark is extracted from the specific positions of Fig. 2 (g) based on LSB modification as mentioned above. The extracted watermark is shown in Fig. 2 (h). As shown in Fig. 2, the embedded watermark is [1 1 0 0] and extracted watermark is [0 0 1 1]. The watermarks are totally different because of the rounding errors.



118	115	114	112	112	115	114	118
115	117	111	118	118	120	117	117
114	117	111	118	117	118	116	118
111	111	114	115	113	117	119	121
115	116	112	113	117	115	118	122
113	110	111	112	117	121	122	119
110	115	115	119	118	123	123	118
120	120	118	119	120	119	126	123

(a) Intensity values of the original 8×8 input image

932,50	-16,26	3,77	3,16	0,00	-0,36	-2,65	111,83
-9,17	6,48	1,32	-0,05	3,12	-2,62	2,08	-3,27
6,91	5,02	0,15	1,11	-1,33	1,15	-0,59	-0,10
-6,64	-1,60	2,33	2,28	-0,74	-0,43	3,01	2,20
0,50	3,14	7,07	-4,65	1,50	0,20	1,40	3,43
-4,12	-2,29	-2,27	-2,45	-2,89	2,37	2,80	-0,98
-0,97	1,14	3,41	1,37	1,36	0,15	0,85	1,74
0,97	3,70	-0,01	-1,01	1,12	-2,96	-2,10	-1,13

(b) DCT transformed image

		0	0				
	1						
1							

(c) The binary watermark

932,50	-16,26	3,77	3,16	0,00	-0,36	-2,65	111,83
-9,17	7,48	1,32	-0,05	3,12	-2,62	2,08	-3,27
7,91	5,02	0,15	1,11	-1,33	1,15	-0,59	-0,10
-6,64	-1,60	2,33	2,28	-0,74	-0,43	3,01	2,20
0,50	3,14	7,07	-4,65	1,50	0,20	1,40	3,43
-4,12	-2,29	-2,27	-2,45	-2,89	2,37	2,80	-0,98
-0,97	1,14	3,41	1,37	1,36	0,15	0,85	1,74
0,97	3,70	-0,01	-1,01	1,12	-2,96	-2,10	-1,13

(d) Watermark embedding in DCT domain

118,40	115,37	114,30	112,21	112,12	115,03	113,96	117,92
115,27	117,24	111,18	118,11	118,03	119,95	116,89	116,86
114,07	117,05	111,01	117,96	116,91	117,86	115,82	117,80
110,88	110,88	113,86	114,85	112,83	116,81	118,80	120,79
114,79	115,80	111,81	112,83	116,85	114,86	117,88	121,88
112,80	109,82	110,86	111,91	116,96	121,01	122,05	119,07
109,86	114,89	114,95	119,03	118,11	123,18	123,24	118,27
119,92	119,96	118,03	119,12	120,21	119,30	126,37	123,40

(e) IDCT transformed watermarked image

118	115	114	112	112	115	114	118
115	117	111	118	118	120	117	117
114	117	111	118	117	118	116	118
111	111	114	115	113	117	119	121
115	116	112	113	117	115	118	122
113	110	111	112	117	121	122	119
110	115	115	119	118	123	123	118
120	120	118	119	120	119	126	123

(f) Rounded Watermarked image

932,50	-16,26	3,77	3,16	0,00	-0,36	-2,65	-3,17
-9,17	6,48	1,32	-0,05	3,12	-2,62	2,08	-3,27
6,91	5,02	0,15	1,11	-1,33	1,15	-0,59	-0,10
-6,64	-1,60	2,33	2,28	-0,74	-0,43	3,01	2,20
0,50	3,14	7,07	-4,65	1,50	0,20	1,40	3,43
-4,12	-2,29	-2,27	-2,45	-2,89	2,37	2,80	-0,98
-0,97	1,14	3,41	1,37	1,36	0,15	0,85	1,74
0,97	3,70	-0,01	-1,01	1,12	-2,96	-2,10	-1,13

(g) DCT transformed watermarked image

		1	1				
	0						
0							

(h) Extracted Watermark

Fig. 2. Watermark embedding and extracting operation by using simple rounding based watermarking technique

To solve the problems above, CSA is used to find the suitable solution in translating real numbers into integers.

The novel CSA based watermark embedding method can be described as shown:

1. Apply DCT to the original input image:

$$I^{DCT} = DCT(I) \tag{3}$$

2. Insert Watermark into the coefficients of I^{DCT} as mentioned above:

$$I^{WDCT} = W \oplus I^{DCT} \tag{4}$$

3. Transform I^{WDCT} by IDCT to obtain I^{WR} :

$$I^{WR} = IDCT(I^{WDCT}) \tag{5}$$

4. Translate the real numbers in I^{WR} into integers by using CSA, and obtain I^{WCSA} .

$$I^{WCSA} = CSA(I^{WR}) \tag{6}$$

The whole procedures of the developed CSA method are illustrated diagrammatically in Fig. 3.

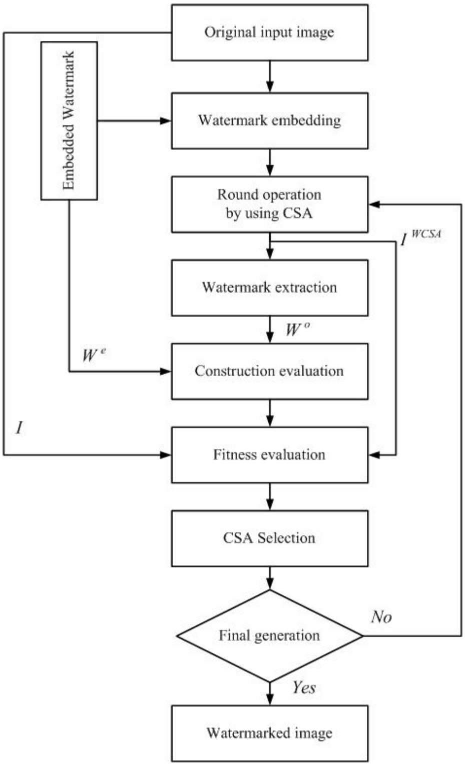
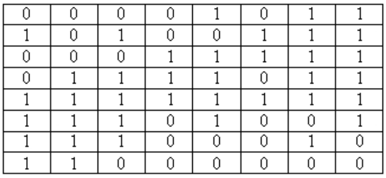


Fig. 3. Block diagram of CSA-based watermarking method

Applying CSA to solve the rounding problem, an antibody (solution set) Ab represented as $Ab = a_1, a_2, \ldots a_{64}$ consists of 64 binary bits where $a_i (1 \le i \le 64)$ correspond to the pixels shown in Fig. 4. One form of the antibody called as translation map is used for rounding operation. An example of the antibody Ab_1 shown in Fig. 4 is represented as:

$$Ab_1 = [0\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0]$$



0	0	0	0	1	0	1	1
1	0	1	0	0	1	1	1
0	0	0	1	1	1	1	1
0	1	1	1	1	0	1	1
1	1	1	1	1	1	1	1
1	1	1	0	1	0	0	1
1	1	1	0	0	0	1	0
1	1	0	0	0	0	0	0

Fig. 4. The solution set of CSA, Ab_1 (the translation map)

CSA algorithm tries to find the best solution on the basis of 8×8 block size. Each block has its own solution given by $I^{DCT}_{(m,n)}$, where $I^{DCT} = \bigcup_{m=1}^{M/8} \bigcup_{n=1}^{N/8} I^{DCT}_{(m,n)}$.

For each block (m,n) in I, the resulting 64 (8×8) DCT bands $I^{DCT}_{(m,n)}$ can be represented by $I^{DCT}_{(m,n)} = \bigcup_{i=1}^{8} \bigcup_{j=1}^{8} I^{DCT}_{(m,n)}(i,j)$.

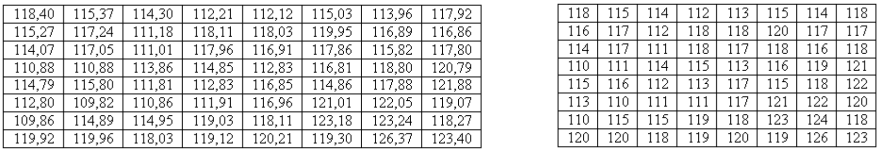
The binary elements of the translation map are used to represent the policy in translating real numbers into integers. When $I^{DCT}_{(m,n)}(i,j)$ is a real number, an integer ${}^{*}I^{DCT}_{(m,n)}(i,j)$ will be obtained using the rules given below:

1) ${}^{*}I^{DCT}_{(m,n)}(i,j) = Trunc\left(I^{DCT}_{(m,n)}(i,j)\right) + 1$, if the (i,j). element of the translation map is "1"

2) ${}^{*}I^{DCT}_{(m,n)}(i,j) = Trunc\left(I^{DCT}_{(m,n)}(i,j)\right)$, if the (i,j). element of the translation map is "0"

The function of $Trunc\left(I^{DCT}_{(m,n)}(i,j)\right)$ denotes the integer part of $I^{DCT}_{(m,n)}$.

An example is illustrated in Fig. 5. Fig. 5(a) shows an image where all pixels are real number. Fig. 5(b) is obtained by translating the real numbers into integer by using the translation map shown in Fig. 4.



118,40	115,37	114,30	112,21	112,12	115,03	113,96	117,92
115,27	117,24	111,18	118,11	118,03	119,95	116,89	116,86
114,07	117,05	111,01	117,96	116,91	117,86	115,82	117,80
110,88	110,88	113,86	114,85	112,83	116,81	118,80	120,79
114,79	115,80	111,81	112,83	116,85	114,86	117,88	121,88
112,80	109,82	110,86	111,91	116,96	121,01	122,05	119,07
109,86	114,89	114,95	119,03	118,11	123,18	123,24	118,27
119,92	119,96	118,03	119,12	120,21	119,30	126,37	123,40

(a) IDCT transformed watermarked image

118	115	114	112	113	115	114	118
116	117	112	118	118	120	117	117
114	117	111	118	117	118	116	118
110	111	114	115	113	116	119	121
115	116	112	113	117	115	118	122
113	110	111	111	117	121	122	120
110	115	115	119	118	123	124	118
120	120	118	119	120	119	126	123

(b) Watermarked image rounded by CSA

Fig. 5. An example of translation operation using CSA

In this paper, CSA uses a constraint function as given below:

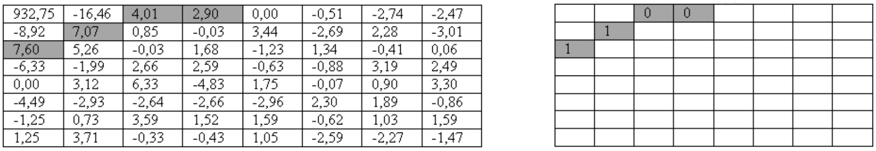
$$C = \sum_{i=1}^{4} \left| Watermark_i^O - Watermark_i^E \right| = 0 \tag{7}$$

where, $Watermark^O$ and $Watermark^E$ are the embedded and extracted watermarks of each blocks, respectively. Fitness function used by CSA is given below;

$$F(gen) = \sum_{i=1}^{64} \left| Watermarked\_image\_block_i - Original\_image\_block_i \right| \qquad (8)$$

where, *gen* represents the evaluation number.

By the constraint function, extracted watermark is obtained exactly the same as the embedded watermark shown in Fig. 6. In addition, optimum watermarked image transparency is also achieved.



932,75	-16,46	4,01	2,90	0,00	-0,51	-2,74	-2,47
-8,92	7,07	0,85	-0,03	3,44	-2,69	2,28	-3,01
7,60	5,26	-0,03	1,68	-1,23	1,34	-0,41	0,06
-6,33	-1,99	2,66	2,59	-0,63	-0,88	3,19	2,49
0,00	3,12	6,33	-4,83	1,75	-0,07	0,90	3,30
-4,49	-2,93	-2,64	-2,66	-2,96	2,30	1,89	-0,86
-1,25	0,73	3,59	1,52	1,59	-0,62	1,03	1,59
1,25	3,71	-0,33	-0,43	1,05	-2,59	-2,27	-1,47

(a) DCT transformed watermarked image of Fig. 5(b)

		0	0				
	1						
1							

(b) Extracted watermark

Fig. 6. Watermark extraction operation

4 Simulation Results

To examine the effect of the control parameter values of CSA, the simulations are applied on a particular block with different population sizes and mutation rates. Then the mean and standard deviation of fitness values obtained for each 30 runs are given in Table 1. As seen from the results, while the population size increases, better fitness values are obtained. In the simulation, the best fitness values are obtained for the mutation rate of 0.03. The obtained relatively smaller standard deviation values confirm the stability of the algorithm. Also the results show that the dependence of CSA algorithm to the control parameter values is fairly low.

Table 1. The effect of the mutation ratio and population size of CSA based novel method on a single block

		Mutation Rate									
		0.01		**0.02**		**0.03**		**0.04**		**0.05**	
		Mean	*Std.*	Mean	*Std.*	Mean	*Std.*	Mean	*Std.*	Mean	*Std.*
Population Size	**40**	16,73	*4,14*	15,47	*3,93*	14,63	*3,34*	15,47	*2,97*	16,03	*2,51*
	60	15	*2,98*	13	*3,50*	12,73	*3,72*	13,17	*3,71*	13,73	*2,83*
	80	12,80	*3,91*	12,27	*3,69*	11,17	*3,67*	11,30	*3,45*	11,60	*2,92*
	100	10,87	*4,30*	11,03	*3,51*	9,17	*3,74*	9,53	*3,55*	9,87	*3,08*

The performance comparison of CSA and GA in terms of convergence speed for the rounding problem on a particular block is shown in Fig 7. The averaged fitness values for both algorithms at each 30 runs represent that CSA is better than GA for this problem. The control parameter values of CSA are chosen, 0.03 for mutation rate, and 0.1 for multiplying factor. And the control parameter values of GA are chosen 0.01 for mutation rate, 0.7 for crossover rate. Also for both algorithm the maximum

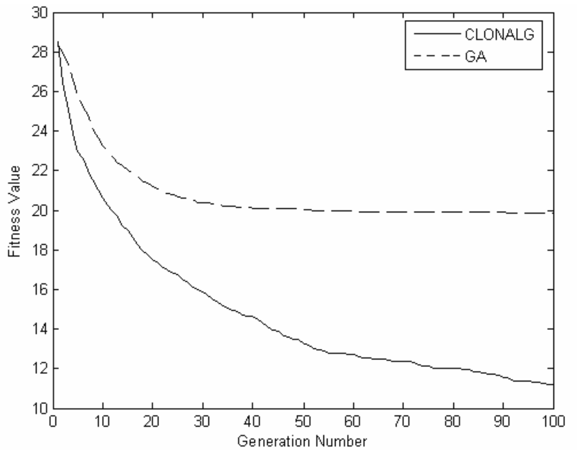
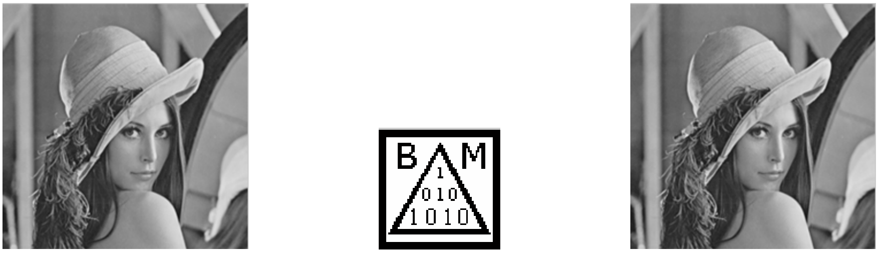


Fig. 7. Averaged fitness value's of the fitness function over generations obtained by CLONALG and GA on a particular block of 30 runs



(a) (b) (c)

Fig. 8. (a) Original Image Lena, (b) Watermark Image, (c) Watermarked Image Lena

generation number and the population size is taken as 100, 80, respectively. As seen from the Fig. 7. CSA has produced better performance than GA.

The performance of the proposed method is evaluated by using Lena image with size 256×256 as the original image, which is shown in Fig. 8 (a). The embedded 64×64 binary watermark image is illustrated in Fig. 8 (b). Hence, the number of bits to be embedded in one 8×8 block is 4. The watermarked image is given Fig. 8 (c).

The comparison of simple rounding, CSA and GA watermarking methods are shown in Table 2. The PSNR values of original and watermarked images are calculated as given below:

$$PSNR = 10\times\log_{10}\left(\frac{255^2}{\frac{1}{N\times N}\sum_{i=1}^{N}\sum_{j=1}^{N}\left[I(i,j)-I^{WCSA}(i,j)\right]^2}\right) \tag{9}$$

where, $I(i,j)$ and $I^{WCSA}(i,j)$ are the pixel values at position (i,j) of the original input image I and watermarked image I^{WCSA}, respectively.

The Normalized Correlation, NC of the extracted watermark is calculated by using the equation below:

$$NC = \frac{\sum_{i=1}^{N} \sum_{J=1}^{N} W^{e}(i,j) \times W^{o}(i,j)}{\sum_{i=1}^{N} \sum_{J=1}^{N} \left[W^{e}(i,j)\right]^{2}} \tag{10}$$

where, $W^e(i,j)$ and $W^o(i,j)$ denote the pixel values at position (i,j) of the embedded watermark W^e and extracted watermark W^o, respectively.

Table 2. Comparison of the simple rounding, CSA based and GA based methods

	By simple rounding	By novel CSA	By GA
PSNR values of original and watermarked images	63,4	57,51	54,65
NC values of embedded and extracted watermark	0,513	1	1

Fig. 9 (a), (b) are the extracted watermarks obtained by using simple rounding and the proposed novel watermarking methods, respectively.

B
M
1
010
1010
(a)
(b)

Fig. 9. Extracted watermarks (a) using simple rounding, (b) using CSA-based method

5 Conclusion

In this paper, we achieve a novel fragile watermarking method based on CSA to correct the rounding errors caused by transforming an image from frequency domain to spatial domain. Simple rounding methods can not be so effective as the proposed novel CSA based watermarking method to extract the embedded watermarks due to rounding errors. Simulation results show that by using the novel CSA-based watermarking method, extracted watermark is obtained exactly the same as embedded watermark. The proposed novel CSA based watermarking method removes these rounding errors completely. In addition, by using this novel method optimum watermarked image transparency is achieved, as well. Moreover, the performance comparison of the CSA, GA and simple rounding methods is applied. The simulation results obtained show that CSA has produced better results than GA. In order to enhance transparency, PSNR values of original and watermarked images are needed increasing. Therefore, our further research will be based on to increase the values of these parameters.

References

1. Potdar, V., Han, S., Chang, E.: A Survey of Digital Image Watermarking Techniques. In: Proceedings of the 3rd International IEEE Conference on Industrial Informatics, Perth Western Australia. IEEE Computer Society Press, Los Alamitos (2005)
2. Pan, J.S., Huang, H.C., Jain, L.C. (eds.): Intelligent Watermarking Techniques, vol. 15. World Scientific Publishing Company, Singapore (2004)
3. Lee, S.J., Jung, S.H.: A Survey of Watermarking Techniques Applied to Multimedia. In: ISIE 2001, Pusan, Korea, pp. 272–277 (2001)
4. Lin, E.T., Podilchuk, I., Delp, E.J.: Detection of Image Alterations Using Semi-Fragile Watermarks. In: Proc. of the SPIE Int. Conference on Security and Watermarking of Multimedia Contents II, vol. 3971 (2000)
5. Holland, J.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI (1975)
6. Karaboğa, D., Okdem, S.: A Simple and Global Optimization Algorithm for Engineering Problems: Differential Evolution Algorithm. Elektrik, Turkish Journal of Electrical & Computer Sciences 12(1), 53–60 (2004)
7. Kumsawat, P., et al.: Wavelet-Based Image Watermarking Using the Genetic Algorithm. In: Negoita, M.G., Howlett, R.J., Jain, L.C. (eds.) KES 2004. LNCS (LNAI), vol. 3215, pp. 643–649. Springer, Heidelberg (2004)
8. Kumsawat, P., Attakitmongcol, K., Srikaew, A.: A New Approach for Optimization in Image Watermarking by Using Genetic Algorithms. IEEE Transactions on Signal Processing 53(12), 4707–4719 (2005)
9. Wang, F., et al.: VQ-Based Gray Watermark Hiding Scheme and Genetic Index Assigment. In: Aizawa, K., Nakamura, Y., Satoh, S. (eds.) PCM 2004. LNCS, vol. 3332, pp. 73–80. Springer, Heidelberg (2004)
10. Shieh, C., et al.: Genetic Watermarking Based on Transform-domain Techniques. Pattern Recognition Society 37(3), 555–565 (2004)
11. Chang, Y.L., Sun, K.T, Chen, Y.H.: ART2-Based Genetic Watermarking, Advanced Information Networking and Applications. In: AINA 2005, 19th International Conference, vol. 1, pp. 729–734 (2005)
12. Shih, F.Y., Wu, Y.: Enhancement of Image Watermark Retrieval Based on Genetic Algorithm. Journal of Visual Communication and Image Representation 16, 115–133 (2005)
13. Aslantas, V.: A Singular-value Decomposition-based Image Watermarking using Genetic Algorithm. Int. J. Electron. Commun. (AEU), doi: 10.1016/j.aeue.2007.02.010 (2007)
14. De Castro, L.N., Von Zuben, F.J.: The clonal selection algorithm with engineering applications. In: Workshop Proceedings of GECCO'00, Workshop on Artificial Immune Systems and their Applications, Las Vegas, USA, pp. 36–37 (2000)
15. De Castro, L.N., Von Zuben, F.J.: Learning and optimization using clonal selection principle. IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems 6(3), 239–251 (2001)
16. The Clonal Selection Theory of Acquired Immunity. Cambridge Univ. Press, Cambridge, U.K (1959)
17. De Castro, L.N., Von Zuben, F.J.: Learning and optimization using the clonal selection principle. IEEE Transactions on Evolutionary Computation 6, 239–251(2002)
18. Ada, G.L., Nossal, G.: The clonal selection theory. Scientific American 257, 50–57 (1987)

BeeAIS: Artificial Immune System Security for Nature Inspired, MANET Routing Protocol, BeeAdHoc

Nauman Mazhar and Muddassar Farooq

Centre for Advanced Studies in Engineering,
Islamabad 44000, Pakistan
{naumaz,mfarooq}@case.edu.pk

Abstract. Artificial Immune Systems *(AIS)* offer a relatively novel and promising paradigm to solve the problem of security in Mobile Adhoc Networks *(MANETs)*. In this paper we address the issue of security in the challenging *MANET* environment by developing an *AIS* based security framework to detect misbehavior in a Bio/Nature inspired *MANET* routing protocol, *BeeAdHoc*. To the best of our knowledge, this is the first attempt to provide AIS based protection in the Bio/Nature inspired domain of MANET routing. We designed and developed a security framework, *BeeAIS*, in the network simulator ns-2. We simulated a number of routing attacks to verify that the *AIS* based security system can counter all of them. These attacks, however, were successful in a MANET running the original *BeeAdHoc* protocol. We also compared our AIS based system with a cryptographic security system, *BeeSec*, developed earlier for *BeeAdHoc*. The results of our extensive experiments clearly indicate the effectiveness of the *AIS* to provide a similar security level as that of the cryptographic solution, but at significantly lower energy and communication cost. The efficient utilization of constrained bandwidth and battery is a key requirement in MANET routing.

1 Introduction

Security in MANETs has fast become an active area of research in recent past. Several security solutions have been proposed for classical MANET routing protocols: major ones being *ARIADNE* [4], which uses symmetric cryptography to secure Dynamic Source Routing *(DSR)* protocol [5], and Secure Ad-Hoc On-demand Distance Vector *(SAODV)* [14], that utilizes asymmetric cryptography for securing Ad-Hoc On-demand Distance Vector *(AODV)* routing protocol [7]. Biological systems have inherent properties of self organization, adaptivity, scalability, robustness and distribution, which are necessary for efficient routing in the challenging MANET environment. Research in this direction has resulted in developing state of the art Bio/Nature inspired routing protocols, *AntHocNet* [2], *Termite* [8] and *BeeAdHoc* [10]. However, security in Nature inspired routing protocols is still an open issue. Widespread acceptance and adoption of these

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 370–381, 2007.

protocols in real world wireless networks would not be possible until their security aspects have thoroughly been investigated. The authors of [12][13] propose two security models for *BeeHive* [11], a Bio/Nature inspired routing protocol for fixed networks. For Nature inspired MANET routing protocols, the security vulnerabilities of *BeeAdHoc* have been studied in [6] and a security framework *(BeeSec)* based on digital signature authentication has been proposed. The authors have demonstrated that it successfully counters a number of attacks on *BeeAdHoc*. Misbehavior detection of a classical routing protocol, *DSR*, using AIS, was proposed in [9]. We in this paper present the design and implementation of a security framework by utilizing the principles of Artificial Immune Systems [3], for misbehavior detection in a Nature inspired MANET protocol, *BeeAdHoc*. To the best of our knowledge, this is the first attempt to secure a Bio/Nature inspired MANET routing protocol using the principles of AIS. The major contributions of the work proposed in this paper are:

- Development of an AIS based security framework, *BeeAIS*, for *BeeAdHoc* protocol and its implementation in ns-2.
- Comprehensive evaluation of the security features of three protocols: *BeeAdHoc*, *BeeSec* and *BeeAIS*. We developed an attacker framework in ns-2 to launch a number of routing attacks and then analyzed their impact.
- The simulation results indicate that the inclusion of our AIS based security framework in *BeeAdHoc* not only successfully counters the routing attacks but also provides superior performance compared to cryptographic solution.

The rest of the paper is organized as follows. Section 2 describes the related work in securing network routing protocols using the AIS approach. Section 3 introduces briefly the *BeeAdHoc* and *BeeSec* protocols. Then in Section 4, we discuss in detail the design of our proposed AIS security framework, *BeeAIS*, along with its implementation in ns-2. Section 5 describes the attacker framework used to launch routing attacks on three protocols: *BeeAdHoc*, *BeeSec* and *BeeAIS*. We compare the security of *BeeAIS* with that of *BeeAdHoc* and *BeeSec*. Our results clearly demonstrate that *BeeAIS* is able to provide the same security level as that of *BeeSec* but it has significantly lower energy and control overhead. Moreover, it does not have the greater processing complexity of asymmetric cryptography. Finally, we conclude the paper with an outlook to our future research.

2 AIS Related Work

AISs [3] have been extensively used for anomaly detection in communication networks. An interested reader can refer to [1] for a comprehensive review. In MANETs, the authors of [9] developed an AIS based security system for misbehavior detection in *DSR* protocol. They successfully detected the packet dropping attacks; non-forwarding of route requests or data packets. The system has an initial *learning* phase (200 seconds maximum) when mobile nodes collect self antigens that consist of sequences of normal *DSR* protocol events and map them to a binary antigenic representation of four genes each. The genes are defined by the designer so that they can encode the correct functioning of the *DSR* protocol. An antigen is collected every 10 seconds and it records the frequency of the normal protocol events. At the end of the learning phase, *negative selection* [3] produces a set of detectors that are used to match and detect non-self antigens in the subsequent *detection and classification* phase. If the detectors match antigens of a node and the probability of that node being a suspicious one is above a threshold, the node is classified as misbehaving. The system in its present form does not cater for fabrication and tampering attacks. Moreover, the system requires significantly large amount of time in order to finally classify a node as misbehaving.

The security of a Bio/Nature inspired routing protocol for fixed networks, *BeeHive* [11], by using the principles of AIS, was proposed in [12]. The system uses a combination of real, symbolic and binary hamming shape spaces [3] to encode antigens and antibodies. The operation of *BeeHive* algorithm requires an *initialization* phase (30 seconds) even before the AIS learning could start. It is followed by the *learning* (50 seconds) and *protection* phases to respectively learn the *BeeHive* normal behavior and detect the routing attacks. Negative selection is used for generation of detectors. The system is capable of detecting malicious nodes that try to artificially change the network topology or modify the route quality to divert the network traffic. Denial of Service *(DOS)* attacks can be detected but the system does not provide protection against dropping attacks.

3 BeeAdHoc and BeeSec Protocols

BeeAdHoc is a source routing protocol, which routes data based on the foraging principles of honey bees [10]. When route to a destination is needed, a *forward scout* is broadcast for route discovery. All receiving nodes append their addresses to the source route, until the scout arrives at the destination node. The destination node then reverses the source route and unicasts the scout back as a *backward scout* to the source node, where the discovered route is advertised to *foragers*. Foragers use the path to transport data to the destination. On their journey, foragers collect the routing information to evaluate the *dance number*, which represents the quality of the path traversed. The foragers have a higher probability of using an advertised path if its quality is higher.

BeeAdHoc Vulnerabilities. An attacker can exploit *BeeAdHoc* agents to launch *fabrication*, *tampering* and *denial of service (DOS)* attacks. When a scout or forager is transmitted an intermediate node can partially modify its source route or insert a completely new route, spoof source/destination addresses and then retransmit the packet claiming to be from some other node. These packets cause installation of forged routes in nodes that allows an attacker to divert the traffic on a path of its own choice. A malicious node can also modify the routing information to make a certain path desirable or undesirable. An attacker can thus sniff packets or prevent traffic to pass through itself to conserve battery. It can also deplete the battery of another node or launch a *DOS* attack by having a large number of packets sent to a particular node.

BeeSec is a security solution for *BeeAdHoc*, which utilizes public-key cryptograpy to protect its messages [6]. The protocol provides for authenticating packet header fields in scouts and foragers to counter fabrication and tampering attacks. A sending node computes digital signatures on the header fields such as *source address, packet ID, destination address, source route, routing information*, using its private key. The receiving node can then use the public key of the sending node to verify that the message did originate from the sender and that the message contents have not been tampered or forged by the nodes it visited.

Here we provide only brief descriptions of *BeeAdHoc* and *BeeSec* protocols. Interested readers can find details in [10] and [6], respectively.

4 BeeAIS: AIS Based Security Model

The proposed AIS security model for *BeeAdHoc* protocol operates on the principle of anomaly detection. It learns the normal behavior of the system and then monitors the system for occurrences of abnormal patterns. The system, therefore, has the ability to detect previously unknown attacks.

4.1 Antigen-Antibody Representations

An AIS requires mapping of the immune system components and quantitative description of interactions between antigens and antibodies [3]. Since, a malicious node can launch attacks on *BeeAdHoc* by manipulating specific packet header fields, we decided to encode and represent the antigens and antibodies with system specific attributes expected to change during the attack.

An antigen is represented as $Ag = [Ag_1, Ag_2, ..., Ag_L]$ and an antibody as $Ab = [Ab_1, Ab_2, ..., Ab_L]$. Both are attribute strings of length L in a shape space S. Now Ag and Ab can be regarded as points in this L-dimensional shape-space $Ag \in S^L$ and $Ab \in S^L$. These strings may comprise real values, integers, symbols or bits. Our proposed system operates in the binary hamming shape space with string length $L = 52$.

Antigen Formats. We modelled three types of antigens: one *scout antigen* and two *forager antigens, type-I and type-II.* The scout antigen detects abnormalities in both the forward and backward scouts. However, two separate forager antigens, type-I and type-II, are required to detect tampering of forager source route and routing information respectively. Figure 1(a) shows the antigen formats. All antigens are 52 bit long bitstrings ($L = 52$) with four genes each, having lengths *16,16,4 and 16* bits respectively.

Detectors. Detectors or antibodies are also 52 bit long bitstrings, which are encoded with the same type of genes as for antigens. To create a detector, each gene value is generated randomly within the specified range as shown in Figure 1(b). Genes are then concatenated to form the complete detector bitstring.

Matching Function. The interaction between Ag and Ab can be evaluated as a distance function, *euclidean, manhattan or hamming.* Their affinity (or matching) then becomes proportional to the distance D between them. To quantify detection, the authors of [3] have proposed the concept of a *recognition region.* An antigen is considered detected if it lies within the *recognition region* of volume V_ϵ around the antibody; ϵ is the radius of the recognition region *(cross reactivity threshold).* In other words, considering similarity measures, an antigen is considered detected if $D \leq \epsilon$. In our system, affinity is measured using hamming distance, with cross reactivity threshold $\epsilon (0 < \epsilon < L)$, where $L = 52$.

Gene 1	Gene 2	Gene 3	Gene 4
16 bits 51... ...36	16 bits 35... ...20	4 bits 19 ... 16	16 bits 15... ...0

Scout Antigen :

Scout Source	Scout ID	Route Length	Previous Hop Address

Forager Antigen (TYPE-I) :

Forager Source	Forager Destination	Route Length	Previous Hop Address

Forager Antigen (TYPE-II) :

Forager Source	Forager Destination	Route Length	Route Information

(a) Antigen format (52 bits long)

52 bits long Antibody (detector) :

Gene 1	Gene 2	Gene 3	Gene 4

Gene 1	16 bit source address	⟨random integer⟩ mod 65536
Gene 2	16 bit destination address or scout ID	⟨random integer⟩ mod 65536
Gene 3	4 bit route length	⟨random integer⟩ mod 16
Gene 4	16 bit route information or previous hop address	⟨random integer⟩ mod 65536

(b) Genes arrangement and Antibody format

Algorithm 1 : Learning Phase

```
for all ( received Scout OR Forager at each node ) do
    if ( Learning Phase AND Scout ) then
        if ( Forward Scout AND already seen ) then
            drop Forward Scout and exit
        end if
        get Scout header fields ⟨source, ID, route length, previous hop⟩
        form Scout Self Antigen bitstring
        determine Hamming Distance of Scout Self Antigen
        if ( Self Antigen matches previous collected Self Antigens ) then
            drop Scout Self Antigen
        else
            store Scout Self Antigen in Scout Self Antigen List for this node
        end if
    end if

    else if ( Learning Phase AND Forager ) then
        get $Hops_{forager}$ number of Forager hops uptil now
        determine Forager $JourneyTime_{learned}$
        compute moving average of $JourneyTime_{learned}$ over 5 Foragers
        store $Hops_{forager}$ vs $JourneyTime_{learned}$ for this node
        get Forager header fields ⟨source, dest, route length, previous hop⟩
        form a Forager Self Antigen TYPE-I bitstring
        get Forager header fields ⟨source, dest, route length, route info⟩
        form a Forager Self Antigen TYPE-II bitstring
        if ( Self Antigen matches previous collected Self Antigen ) then
            drop matching Forager Self Antigen
        else
            store TYPE-I Self Antigen in TYPE-I Forager Self Antigen List
            for this node
            store TYPE-II Self Antigen in TYPE-II Forager Self Antigen List
            for this node
        end if
    end if
end for
```

Fig. 1. Antigen/Antibody formats and BeeAIS learning phase algorithm

4.2 BeeAIS Operation

The *BeeAIS* framework consists of two phases: learning and protection.

Learning Phase. In this phase, the network is assumed to be free of non-self antigens and *BeeAIS* defines 'self' by profiling the normal behavior of the monitored system. *BeeAIS* has a *learning phase* of 50 seconds. Figure 1 shows the algorithm. Each node in the network gathers its own set of self antigens. When a scout (*forward* or *backward*) is received, one scout antigen is formed. In case of receiving a forager, both the type-I and type-II forager antigens are created. A

Algorithm 2 : Dectector Generation

```
if ( current time == end of Learning Phase ) then
  while ( Scout Detectors < $Num_{detectors}$ ) do
    randomly generate all four Gene values
    form an Immature Detector
    retrieve Scout Self Antigen List for this node
    if ( Detector matches a Scout Self Antigen ) then
      discard Immature Detector
    else
      store Detector in Scout Detector List for node
    end if
  end while
  while ( Forager Dectectors TYPE-I < $Num_{detectors}$ ) do
    randomly generate all four Gene values
    form an Immature Detector
    retrieve Forager Self Antigen List TYPE-I for this node
    if ( Detector matches a Forager Self Antigen ) then
      discard Immature Detector
    else
      store Detector in Forager Detector List TYPE-I
      for this node
    end if
  end while
  while ( Forager Dectectors TYPE-II < $Num_{detectors}$ ) do
    randomly generate all four Gene values
    form an Immature Detector
    retrieve Forager Self Antigen List TYPE-II for this node
    if ( Detector matches a Forager Self Antigen ) then
      discard Immature Detector
    else
      store Detector in Forager Detector List TYPE-II
      for this node
    end if
  end while
end if
```

Algorithm 3 : Protection Phase

```
for all ( received Scout OR Forager at each node ) do
  if ( Protection Phase AND Scout ) then
    if ( Forward Scout AND already seen ) then
      drop Forward Scout and exit
    end if
    get Scout header fields ⟨source, ID, route length, previous hop⟩
    form a Scout Self Antigen bitstring
    determine Hamming Distance of Scout Self Antigen
    retrieve Scout Dectector List for this node
    if ( Self Antigen matches any Scout Dectector ) then
      drop Scout
    end if
  end if

  else if ( Protection Phase AND Forager ) then
    get Forager header fields ⟨source, dest, route length, previous hop⟩
    form a Forager Self Antigen TYPE-I bitstring
    determine Hamming Distance of Forager Self Antigen
    retrieve Forager Dectector List TYPE-I for this node
    if ( Self Antigen matches any Forager Dectector TYPE-I ) then
      drop Forager
    else
      get Forager header fields ⟨source, dest, route length, route info⟩
      form a Forager Self Antigen TYPE-II bitstring
      determine Hamming Distance of Forager Self Antigen
      retrieve Forager Dectector List TYPE-II for this node
      if ( Self Antigen matches any Forager Dectector TYPE-II ) then
        get $Hops_{forager}$ number of Forager hops uptil now
        replace route info in Forager header with $RouteInfo_{forager}$
        obtained from $JourneyTime_{learned}$ against $Hops_{forager}$
      end if
    end if
  end if
end for
```

Fig. 2. Detector generation (negative selection) and protection phase algorithms

node matches the newly created self antigen with the previously collected ones and maintains a list of unique self antigens. At the end of *learning phase*, detector patterns (antibodies) are generated using the *negative selection* algorithm of the *thymus model* [3]. Figure 2 shows the detector generation algorithm. Three detectors sets, each having the number of detectors ($Num_{detectors}$) as 50, are generated at each node; there is one scout detector set and two forager type-I/type-II detector sets. Since each node encounters a different set of traffic during its *learning phase*, the detector sets at each node are unique. As a result, our AIS can operate in a distributed manner.

While forming a type-II forager self antigen, a node also builds a list of tuples of the type $\{Hops_{forager}, JourneyTime_{learned}\}$. $Hops_{forager}$ are the hops between the source node and the current node. Similarly, forager journey time ($JourneyTime_{learned}$) is calculated as follows:

$$JourneyTime_{learned} = Time_{current} - Time_{start} \tag{1}$$

where $Time_{current}$ and $Time_{start}$ are the current time and the forager launch time at the source node respectively. The $JourneyTime_{learned}$ is maintained as the moving average of forager journey times over a window of last five foragers.

Protection Phase. In this phase, detectors are used for matching subsequent profiled patterns of the monitored system. Each node collects antigens (scout antigens and forager antigens of type-I/type-II) and matches them with its own respective detector sets (scout detector set and forager detector sets of type-I/type-II). If a match is found, it indicates detection of a non-self antigen or occurrence of an anomaly in the system. Figure 2 shows the algorithm for the *protection phase*. In the case of foragers, our AIS system first matches the type-I antigen and drops the forager in case of a match, indicating tampered source route. If forager is not dropped, then type-II antigen is matched. A match here indicates tampered routing information. The routing information is then corrected by replacing it with the expected correct route information value ($RouteInfo_{forager}$), which is computed from the forager learned journey time ($JourneyTime_{learned}$) collected during the *learning phase*, as:

$$RouteInfo_{forager} = Time_{current} - JourneyTime_{learned} \tag{2}$$

$JourneyTime_{learned}$ is obtained from the $\{Hops_{forager}, JourneyTime_{learned}\}$ tuples generated by the node during *learning phase*. The value is selected on the basis of the number of hops in between the source node and the current node.

5 Attack Simulations and Performance Evaluation

We realized the proposed security model, *BeeAIS*, in network simulator ns-2 and performed simulations to compare its security and performance characteristics with that of *BeeAdHoc* and *BeeSec*. We also developed an attacker framework in ns-2 to generate different types of routing attacks, discussed in Section 3, alongwith a traffic scope to determine the impact of these attacks.

5.1 Simulation Topology

Simulations were performed using a grid of 49 nodes (Figure 3) covering a rectangular area of operation ($1400 \times 1100m^2$). This is considered a relatively large MANET. Four different attack scenarios were selected; each scenario having one *source* node and one *destination* node, with constant bit rate *(CBR)* traffic from the source to destination. The simulation time for each run is 1000 seconds and the reported results are an average over five independent runs. We decided to have immobile nodes to make it easier to systematically analyze the impact of attacks. The analysis, however, is equally applicable to a scenario of mobile nodes in a network that is smaller or larger than our 49 node MANET.

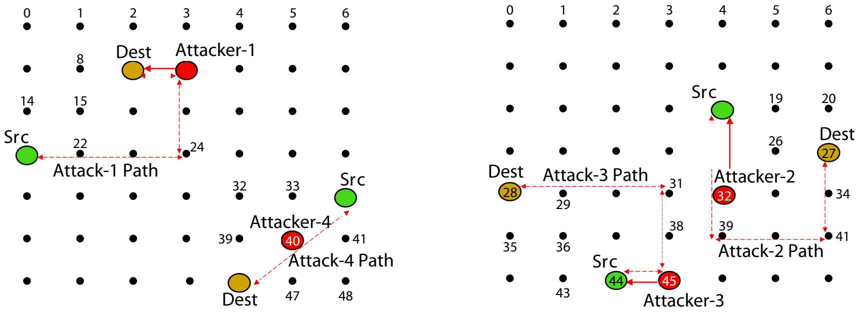
5.2 Performance Metrics

Performance evaluation of the algorithms was done using the following metrics:

Average Throughput (T_{av})**.** *The number of data bits delivered to the application layer at the destination node in a unit interval of time.*

Packet Delivery Ratio (PDR)**.** *The ratio of data packets successfully delivered to destination nodes and total number of packets generated for those destinations.*

Average Latency ($Latency_{av}$)**.** *The average difference in time when a packet is generated at the source and when it got delivered to the destination.*



(a) Attack Scenarios; Attack-1, Attack-4

(b) Attack Scenarios; Attack-2, Attack-3

Fig. 3. Network topology and attack scenarios

Energy per user data (EPD)**.** *Energy consumed in transporting one kilobyte of data to its destination node.*

Transmission efficiency ($Trans_{eff}$)**.** *The number of data bytes delivered to the application at destination nodes at the cost of a unit control byte.*

Average Number of Hops (Hop_{av})**.** *The average number of hops for all the paths traversed by data packets.*

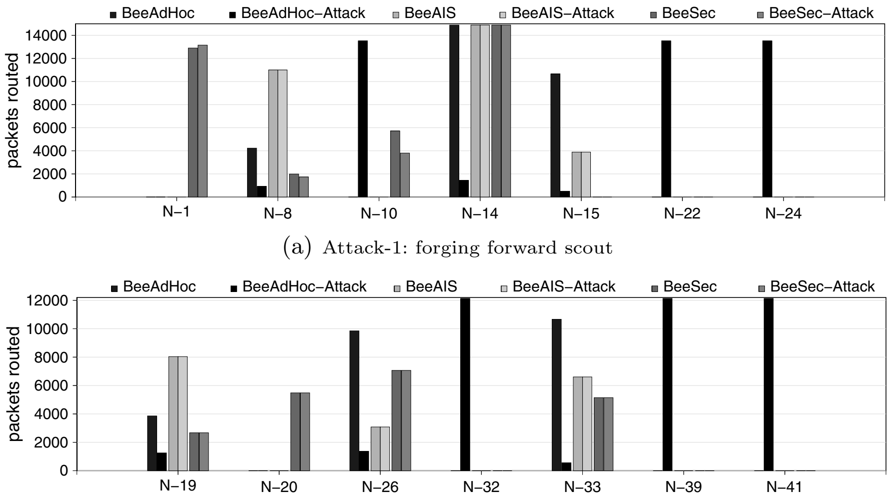
Control Overhead ($CtlBytes$)**.** *Total number of control bytes transmitted by all nodes in the network.*

5.3 Simulating Routing Attacks

The attacks are launched during *protection* phase of *BeeAIS*. Before that, a 50 seconds *learning* phase is allowed to complete so that each node collects its *self antigens* and generates *detectors*, which are then used to match with the traffic during *protection* phase. The attacks are launched at time $t = 51$ seconds.

Attack-1: Forging Forward Scout. We implemented this attack using *Node-21* as the source and *Node-9* as destination, Figure 3(a). Malicious *Node-10* launched forged forward scouts. In *BeeAdHoc* protocol, the forged scouts were returned by the destination *(Node-9)* and caused installation of the path *21-22-23-24-17-10-9* at the source *(Node-21)*. We can see the result of the attack in Figure 4(a). Malicious *Node-10* was successfully able to divert data packets towards itself. The traffic through *Nodes 8, 14, 15* significantly decreased while traffic through *Nodes 10, 22, 24* increased. In contrast, the attack had no effect on *BeeSec* and *BeeAIS* routing because both protocols were able to detect and drop the forged scouts.

Attack-2: Forging Backward Scout. In this attack, *Node-18* was taken as the source and *Node-27* as destination, and backward scouts were forged by *Node-32*, Figure 3(b). We can see the result of the successful attack on *BeeAdHoc* in Figure 4(b). Prior to the attack there was no traffic passing through *Node-32*. When the malicious *Node-32* launched forged backward scouts directed at the source *Node-18*, the forged path *18-25-32-39-40-41-34-27* was established. As a



(a) Attack-1: forging forward scout

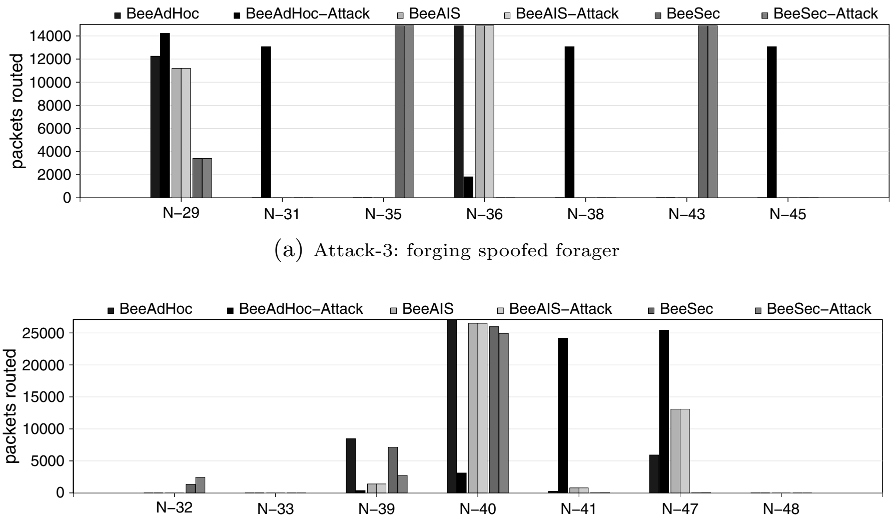
(b) Attack-2: forging backward scout

Fig. 4. Attacks: forging forward scout and forging backward scout

result all subsequent data packets followed the newly forged path, which passes the *Nodes 32, 39, 41*. However, we can see for *BeeSec* and *BeeAIS* protocols that the attack failed to alter the routing behavior.

Attack-3: Forging Spoofed Forager. For this attack, *Node-44* was taken as the source and *Node-28* as destination, Figure 3(b). Attack was launched by *Node-45* by sending spoofed foragers to the source *Node-44*, claiming to be from the destination *Node-28* and carrying the source route *28-29-30-31-38-45-44*. As a result, in *BeeAdHoc* protocol, the forged path *44-45-38-31-30-29-28* was established and subsequent foragers started following the forged path. As can be seen in Figure 5(a), the malicious *Node-45* was successfully able to increase the traffic passing through it. This is indicated by the number of forgares traversing *Nodes 31, 38 45* before and after the attack. On the other hand, *BeeSec* and *BeeAIS* were both able to counter the attack by detecting the forged foragers.

Attack-4: Modifying Forager Route Information. For each received forager exchanged between the source *Node-34* and destination *Node-46*, Figure 3(a), the malicious *Node-40* manipulated the routing information carried in the forager header by artificially increasing the delay value. This resulted in making all the paths through *Node-40* undesirable. In *BeeAdHoc* protocol, the attack was successful and its effect can be seen in Figure 5(b). *Node-40* was able to decrease the traffic passing through it. The traffic then followed other paths through *Nodes 41, 47*. Figure 5(b) also shows that both *BeeSec* and *BeeAIS* protocols remained uneffected by the attack.



(a) Attack-3: forging spoofed forager

(b) Attack-4: modifying forager route information

Fig. 5. Attacks: forging spoofed forager and modifying forager route information

5.4 Performance Comparison

The performance comparison of three protocols *BeeAdHoc*, *BeeSec* and *BeeAIS* has been done with and without attacks. The results are summarized in Table 1. One can see that the AIS based security framework, *BeeAIS*, in each of the four simulation scenarios, had nearly the same performance as that of the original protocol *BeeAdHoc*. *BeeAIS* delivered the same number of data packets as compared to *BeeAdHoc*, with similar delay values and similar energy consumption. Moreover, the protocol suffered negligible degradation under attack and provided security at no additional cost to the system. This is a major advantage when compared to the cryptographic security system *BeeSec*. *BeeSec* did provide the same security level as *BeeAIS* (Section 5.3), but its performance degraded for most of the scenarios when compared to *BeeAdHoc*. Even more important is the fact that the performance of *BeeSec* is significantly inferior as compared to *BeeAIS*. If we average out the results over all four scenarios for *BeeSec* and *BeeAIS* without attacks, we observe that *BeeAIS* has 30% higher average throughput, 25.5% lower latency, 22.5% lower consumed energy, 929.6% higher data transmission efficiency and 90.25% lower control overhead. This is a significant achievement for our AIS based security framework to provide security comparable to asymmetric cryptography but at a significantly smaller energy and bandwidth costs. *BeeSec* transmitted 930.2% more control bytes and consumed 30.6% more battery to provide the same security level.

Table 1. Performance results of *BeeAdHoc*, *BeeSec* and *BeeAIS* with/without attacks (column headings are explained in section 5.2)

$Attack$	$Algorithms$	T_{av} (kb/s)	PDR	$Latency_{av}$ (ms)	EPD (mJ/kB)	$Trans_{eff}$	Hop_{av}	$CtlBytes$
Attack-1	***BeeAdHoc***	**115.120**	**100.00**	**25.377**	**16.226**	**6.739**	**3.989**	**1192832**
	BeeAdHoc(attack)	76.511	98.16	41.443	26.265	2.304	5.790	3497852
	BeeSec	**76.688**	**99.98**	**39.764**	**24.767**	**0.623**	**5.284**	**12911320**
	BeeSec(attack)	75.765	99.98	40.213	25.399	0.611	5.309	13155380
	BeeAIS	**115.660**	**100.00**	**25.288**	**16.171**	**6.774**	**3.976**	**1186624**
	BeeAIS(attack)	115.125	100.00	25.426	16.653	5.910	3.976	1360048
Attack-2	***BeeAdHoc***	**172.887**	**100.00**	**17.673**	**11.336**	**12.207**	**2.797**	**658504**
	BeeAdHoc(attack)	91.123	100.00	39.964	25.832	2.584	6.127	3110664
	BeeSec	**143.079**	**100.00**	**22.589**	**14.147**	**1.185**	**3.060**	**6782428**
	BeeSec(attack)	142.652	100.00	22.731	14.796	1.116	3.060	7202428
	BeeAIS	**187.522**	**100.00**	**17.063**	**10.947**	**12.478**	**2.699**	**644224**
	BeeAIS(attack)	187.168	100.00	17.115	11.254	10.121	2.699	794224
Attack-3	***BeeAdHoc***	**163.667**	**100.00**	**18.513**	**11.869**	**11.391**	**2.930**	**705720**
	BeeAdHoc(attack)	87.777	99.96	36.382	22.962	3.528	5.599	2282308
	BeeSec	**126.497**	**100.00**	**24.046**	**15.063**	**1.105**	**3.255**	**7273564**
	BeeSec(attack)	126.468	100.00	24.058	15.079	1.105	3.255	7272244
	BeeAIS	**163.228**	**100.00**	**19.053**	**12.211**	**10.665**	**3.013**	**753760**
	BeeAIS(attack)	163.204	100.00	19.061	12.225	10.683	3.013	752440
Attack-4	***BeeAdHoc***	**181.701**	**100.00**	**17.485**	**11.207**	**12.129**	**2.766**	**1215020**
	BeeAdHoc(attack)	158.050	100.00	18.630	11.949	11.464	2.951	1285484
	BeeSec	**174.326**	**99.92**	**20.030**	**11.936**	**1.437**	**2.588**	**10276404**
	BeeSec(attack)	182.418	99.59	20.345	11.388	1.515	2.460	9765372
	BeeAIS	**190.733**	**100.00**	**15.972**	**10.250**	**14.559**	**2.535**	**1012236**
	BeeAIS(attack)	190.733	100.00	15.972	10.250	14.559	2.535	1012236

6 Conclusion and Future Work

In this paper we have presented an AIS based security model, *BeeAIS*, for a Bio/Nature inspired MANET routing protocol, *BeeAdHoc*, and compared its security with a cryptographic security framework, *BeeSec*. The AIS was designed and implemented in network simulator ns-2. We also realized in ns-2, an attacker framework that allowed us to launch a number of routing attacks and study their impact. The results of our extensive attack simulations verify that a malicious node can seriously disrupt the routing behavior of *BeeAdHoc* protocol. However, *BeeAIS* can successfully detect the non-self antigens and drop them to counter the attacks. We also find that *BeeAIS* provides this security at no additional control or energy costs to the system. Its performance remains nearly the same as that of its base protocol, *BeeAdHoc*. In comparison, the cryptographic based system, *BeeSec*, also prevents these attacks, but has a high control overhead (930.2% more control bytes) due to the large size of authentication header fields, and 30.6% higher battery consumption. This is in addition to the processing load of asymmetric cryptography. Our work clearly demonstrates that AIS based security has the potential to offer significantly higher performance in MANETs due to its significantly less control, energy and computational cost. The efficient utilization of these resources is a key challenge in MANETs. In future, we intend to extend our basic AIS to learn and adapt to a changing self through the use of *danger signal* and also to counter dropping attacks, which of course is not possible through cryptographic security. This would be a cardinal step in deploying Bio/Nature inspired routing protocols in real world MANETs.

References

1. 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. Springer, Heidelberg (2004)
2. Di Caro, G., Ducatelle, F., Gambardella, L.M.: Anthocnet: An adaptive natureinspired algorithm for routing in mobile ad hoc networks. European Transactions on Telecommunications 16(2), 443–455 (2005)
3. de Castro, L.N., Timmis, J.: Artificial immune systems: a new computational intelligence approach. Springer, Heidelberg (2002)
4. Hu, Y.-C., Perrig, A., Johnson, D.B.: Ariadne: A secure on-demand routing protocol for ad hoc networks. Wireless Networks 11(1-2), 21–38 (2005)
5. Johnson, D.B., Maltz, D.A.: Dynamic source routing in ad hoc wireless networks. In: Imielinski, Korth (eds.) Mobile Computing, pp. 153–181. Kluwer Academic Publishers, Dordrecht (1996)
6. Mazhar, N., Farooq, M.: Vulnerability analysis and security framework (beesec) for nature inspired manet routing protocols. In: Proceedings of GECCO-2007, London, UK, July 2007 (accepted to be published, 2007)
7. Perkins, C., Royer, E.: Ad-hoc on-demand distance vector routing. In: Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications, February 1999, pp. 90–100. IEEE Computer Society Press, Los Alamitos (1999)
8. Roth, M., Wicker, S.: Termite: Ad-hoc networking with stigmergy. In: Proceedings of IEEE GLOBE-COM, San Francisco, USA, December 2003. IEEE Computer Society Press, Los Alamitos (2003)
9. Sarafijanovic, S., Le Boudec, J.Y.: An artificial immune system approach with secondary response for misbehavior detection in mobile ad-hoc networks. IEEE Transactions on Neural Networks 16(5) (September 2005)
10. Wedde, H.F., Farooq, M., Pannenbaecker, T., Vogel, B., Mueller, C., Meth, J., Jeruschkat, R.: Beeadhoc: an energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior. In: GECCO, pp. 153–160 (2005)
11. Wedde, H.F., Farooq, M., Zhang, Y.: Beehive: An efficient fault-tolerant routing algorithm inspired by honey bee behavior. In: Dorigo, M., Birattari, M., Blum, C., Gambardella, L.M., Mondada, F., Stützle, T. (eds.) ANTS 2004. LNCS, vol. 3172, pp. 83–94. Springer, Heidelberg (2004)
12. Wedde, H.F., Timm, C., Farooq, M.: Beehiveais: A simple, efficient, scalable and secure routing framework inspired by artificial immune systems. In: PPSN, pp. 623–632 (2006)
13. Wedde, H.F., Timm, C., Farooq, M.: Beehiveguard: A step towards secure nature inspired routing algorithms. In: Rothlauf, F., Branke, J., Cagnoni, S., Costa, E., Cotta, C., Drechsler, R., Lutton, E., Machado, P., Moore, J.H., Romero, J., Smith, G.D., Squillero, G., Takagi, H. (eds.) EvoWorkshops 2006. LNCS, vol. 3907, pp. 243–254. Springer, Heidelberg (2006)
14. Zapata, M.G.: Secure ad hoc on-demand distance vector (saodv) routing. Internet-Draft, draft-guerrero-manet-saodv-05.txt (February 2005)

A Cultural Immune System for Economic Load Dispatch with Non-smooth Cost Functions

Richard A. Gonçalves[1,3], Carolina P. de Almeida[1,3], Myriam R. Delgado[1],
Elizabeth F. Goldbarg[2] and Marco C. Goldbarg[2]

[1] Federal Technological University of Paraná, Curitiba, PR, Brazil
[2] Federal University of Rio Grande do Norte, RN, Brazil
[3] UNICENTRO, Guarapuava, PR, Brazil

Abstract. This paper presents a novel and efficient method for solving economic load dispatch problems with non-smooth cost functions, by combining an Artificial Immune Systems with Cultural Algorithms. The proposed method, called Cultural Immune System, uses a real coded AIS that is derived from the clonal selection principle with a pure aging operator and hypermutation operators based on Gaussian and Cauchy mutations that are guided by four knowledge sources stored in the belief space of a Cultural Algorithm. The Cultural Immune System has a local search stage that is based on a quasi-simplex technique and several points of self-adaptation. Three test systems with thermal units whose fuel cost function takes into account valve-point loading effects are used to validate the proposed method. These test systems constitute complex constrained optimization problems. Firstly, Cultural Immune System is compared with his non-cultural counterpart (the same AIS without knowledge sources guiding the hypermutation operators). After that both immune-based methods are compared with state-of-the-art algorithms. The results show that the Cultural Immune System is capable of outperforming other state-of-the-art algorithms in solving load dispatch problems with the valve-point effect.

1 Introduction

In the last few years, the use of hybrid methods inspired by Natural Computing [1] has attracted the attention of many researchers, specially the systems in which two or more methodologies are joined to enhance the final model. Some of such hybrid approaches seems to be robust enough to tackle practical hard optimization problems, such as the economic load dispatch (ELD).

The primary objective of the economic load dispatch problem is to determine the optimal quantity of energy that must be generated by each unit to meet the required load demand at minimum operating cost while satisfying system equality and inequality constraints [20]. Traditionally, the cost of each unit is approximated by a quadratic function which can be exactly solved using mathematical programming based on the optimization techniques such as lambda-iteration method, gradient method, and dynamic programming [16]. These methods require incremental fuel cost curves, which should be monotonically increasing, to find a global optimal solution. Unfortunately, the input-output characteristics of generating units are inherently nonlinear and highly

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 382–394, 2007.

non-smooth because of valve-point loadings. Thus, the practical ELD problem with valve-point effects is addressed as a hard optimization problem with equality and inequality constraints that make the problem of finding the global optimum very difficult. Thus, it normally renders classical optimization methods ineffective.

In the absence of exact methods, the stochastic ones (specially the naturally inspired methods such as Artificial Immune Systems [18], Genetic Algorithms [26][17], Particle Swarm [25][20][19] [21] and Differential Evolution [22][23]) have grown in popularity for this practical engineering problem.

This paper combines an Artificial Immune Systems with Cultural Algorithms. The interest in AIS is justified by the Cutello's work [14], that proves the convergence of a class of AIS, particularly those inspired by the clonal selection principle, to the global optimum of an optimization problem. Here we try to improve the already known good performance of AIS by incorporating knowledge sources typical of Cultural Algorithms [2][4][5] [6][7] and a local search phase based on a quasi-simplex technique [17] to an AIS based on the clonal selection algorithm (Clonalg) [10] with an aging operator [11][12][13].

2 Economic Load Dispatch

Economic load dispatch is one of the most important problems to be solved in the operation and planning of a power system [20]. The objective of the economic dispatch problem is to minimize the total fuel cost of thermal power plants subjected to the operating constraints of a power system. The objective function can be formulated as:

$$\begin{array}{c}\text{Minimize F} = \sum_{j=1}^{n} F_j(P_j) \\ \text{subject to} \\ PD = \sum_{j=1}^{n} P_j \text{ and} \\ P_j^{min} \le P_j \le P_j^{max}\end{array} \tag{1}$$

where $F_j(P_j)$ is the fuel cost function of the *jth* generator (in \$/hr), P_j is the power output of the *jth* unit, *n* is the number of generating units in the system, PD is the total power demand, P_j^{min} and P_j^{max} are, respectively, the minimum and maximum power outputs of the *jth* unit. The equality constraint of Equation 1 is called power balance constraint while the inequality constraints are called operational constraints.

The fuel cost function, considering valve-point effects, of each generator is given by a mix of a quadratic approximation and a sinusoidal function [20]:

$$F_j(P_j) = a_j * P_j^2 + b_j * P_j + c_j + |e_j * \sin(f_j * (P_j^{min} - P_j))| \tag{2}$$

where P_j is the power output of the *jth* unit, a_j, b_j, c_j, e_j and f_j are the fuel cost coefficients of the *jth* unit with valve point effects.

3 Cultural Algorithms

Cultural algorithms (CA) are techniques that add domain knowledge to evolutionary computation methods and are derived from the cultural evolution process [4]. They assume that the knowledge related to the problem being solved can be extracted from individuals of a population during the evolutionary process and subsequently used to guide the search.

Cultural algorithms have two main components: the population space, and the belief space [8]. The population space consists of a set of possible solutions to the problem, and can be modeled using any population based technique, such as Evolutionary Programming [8], Particle Swarm [3] and Differential Evolution [6][7].

The belief space is the information repository in which individuals can store their experiences that can be indirectly learned by the other individuals [7]. In cultural algorithms, the information acquired by an individual can be shared with the entire population.

The population space and the belief space are linked through a communication protocol, which states the rules about the individuals that can contribute to the belief space with its experiences (the acceptance function), and the way the belief space can influence the generation of new individuals (the influence function) [6].

The knowledge stored into the belief space can, generally, be divided into five knowledge sources [5][7]: Situational Knowledge, Normative Knowledge, Historical Knowledge, Topographical Knowledge and Domain Knowledge. It is important to note that these knowledge sources were derived from works in cognitive science and semiotics that describe the basic knowledge used by human decision-makers [4] and will be detailed in Section 5.

4 Proposed Artificial Immune System

Immune Algorithms are stochastic algorithms inspired by immunology, immune functions and principles observed in nature [14]. In this paper we are particularly interested in Immune Algorithms inspired by the Clonal Selection Principle, such as the Clonalg [10] and opt-IMMALG [13] which can be classified as Evolutionary Algorithms [9]. This interest is justified by the recent proof that this class of algorithm (under certain conditions) is capable of finding the global optimum of an optimization problem [14].

Clonal Selection Algorithms work with a population of candidate solutions (antibodies), composed of a subset of memory cells (the best ones) and a subset of other good individuals. At each generation a set of best individuals in the population are selected based on their affinity measures (how good they match the antigens). The selected individuals are cloned, giving rise to a temporary population of clones. The clones are submitted to a hypermutation operator, whose rate is proportional (or inversely proportional) to the affinity between the antibody and the antigen (the problem to be solved). From this process a maturated antibody population is generated. Some individuals of this temporary population are selected to be memory cells or to be part of the next population. This whole process is repeated until a termination condition is achieved [10]. Recent works related improved performance by adding an aging operator [12][13][15].

In the Immune System proposed in this paper to solve the Economic Load Dispatch each antibody is a valid combination of power outputs (encoded as real numbers) for the generator units. The affinity of an antibody with the antigen is given by:

$$aff_i = (MaxCost_{gen} - Cost_i)/(MaxCost_{gen} - MinCost_{gen}) \tag{3}$$

where $Cost_i$ is the total fuel cost of a particular configuration (represented by the ith antibody), $MinCost_{gen}$ and $MaxCost_{gen}$ are the maximum and minimum fuel cost of an antibody in generation gen, respectively.

The steps of the implemented Immune System are the same as those presented in Pseudo-Code 2 (see Section 5) with the following modifications: steps 3, 4.8, 4.9 and 4.10 are eliminated of the algorithm and step 4.2 is substituted by "Apply a Hypermutation to the Clones".

Each antibody is composed of n components (power generators) that are initialized at the first generation as $ant_{i,j} = P_i^{min} + U(0,1) * (P_i^{max} - P_i^{min})$, where $ant_{i,j}$ is *jth* component of the *ith* antibody (i.e., the power output of the *jth* unit generator), $U(0,1)$ is a random variable sampled from an uniform distribution in the interval $[0,1]$, P_i^{min} and P_i^{max} are, respectively, the minimum and maximum power outputs of the *jth* unit. This initialization procedure guarantees that no antibody violates the operational constraints of the generator units.

After the evaluation of the initial generation, the Immune System enters its main loop (which represents the affinity maturation stage of the algorithm). This loop is repeated until a termination criterion is satisfied. The termination criterion adopted in this work is a maximum number of generations.

The affinity maturation process begins by cloning the antibodies of the past generation. The most common cloning operators are: static cloning operator (where the number of clones of each antibody is dup, independently of the affinity of the antibody) and the proportional cloning operator (where the number of clones of each antibody is proportional to its affinity) [14]. In this work the static cloning operator is used.

The hypermutation operators are subsequently applied. The implemented algorithm makes use of two hypermutation operators: an Adaptive Gaussian Hypermutation (AGH) and an Adaptive Cauchy Hypermutation (ACH). This hypermutation operators are given by Equation 4.

$$ant_{i,j} = ant_{i,j} + mult * (Cost_i/MinCost_{gen}) * R(0,1) * (P_j^{max} - P_j^{min}) \quad (4)$$

where $ant_{i,j}$ is the *jth* component of the *ith* antibody, $mult$ is an adaptive multiplier given by $mult = e^{-gen/\varphi}$ (where φ is a parameter that dictates the decrease speed of the multiplier), P_j^{max} and P_j^{min} are the maximum and minimum limits of the *jth* generator unit, respectively, $R(0,1)$ is a random number sampled according to a Gaussian or a Cauchy distribution depending on the hypermutation type (AGH or ACH). In both cases the distributions have mean equal to zero and variance equal to one.

The term $Cost_i/MinCost_{gen}$ makes the mutation more intensive in antibodies with a high fuel cost (low affinity) and smooth in antibodies with low fuel cost (high affinity). According to [9] Cauchy random numbers allow relatively coarse-grained steps, while Gaussian random numbers produce fine-grained steps, which in theory is a good balance. The multiplier tries to make the search in the beginning of the evolution intense and smooth at the end of the evolution.

The number of mutations applied to each antibody is given by $M(i) = \frac{e^{-aff_i}}{\rho} * n$, where $M(i)$ is the number of mutations applied to the *ith* antibody, aff_i is the affinity of the *ith* antibody, ρ is a parameter that regulates the number of mutations and *n* is the number of generators.

After hypermutation a quasi-Simplex method is applied as a local search procedure to the best antibodies of the hypermutated clones. At this point the constraints can be violated, which could cause infeasible antibodies. To avoid such violation a repair process is applied to each clone in order to guarantee that the generated antibodies are feasible. So instead of penalizing infeasible antibodies we repair them. The implemented procedure is shown in Pseudo-Code 1.

```
Repeat for each component $j$ of an antibody $i$
  If $ant_{i,j} < P_j^{min}$
    $ant_{i,j} = P_j^{min}$
  Else
    If $ant_{i,j} > P_j^{max}$
      $ant_{i,j} = P_j^{max}$
  End If
End repeat
While $\sum_{i=1}^{n} P_i \neq PD$
  Randomly select a component $j$
  If $\sum_{i=1}^{n} P_i < PD$
    Add an amount to $ant_{i,j}$ that doesn't violate the
    operational constraint and minimize the power balance
    demand violation
  Else
    Subtract an amount from $ant_{i,j}$ that doesn't violate the
    operational constraint and minimize the power balance
    demand violation
End While
```

Pseudo-Code 1: Repair Procedure

After the application of the repair procedure the affinity given by Equation 3 can be used to evaluate the goodness of an antibody.

The next step in the implemented algorithm is the application of the aging operator. In this work the static pure aging operator [12][13] is used. This aging operator eliminates old antibodies in order to maintain the diversity of the population and to avoid the premature convergence. In this operator an antibody is allowed to survive for at most τ_B generations, after this period it is assumed that this antibody corresponds to a local optima and must be eliminated from the population. A clone inherits the age of its parent and is assigned an age equal to zero when it is successfully hypermutated (i.e., when the hypermutation improves the affinity of the antibody).

Finally, the last step of the affinity maturation process is the selection of the antibodies that will compose the next population. The scheme used is a $(\mu + \lambda)$-Selection operator [13] that is applied to parents and hypermutated clones that survived after the aging operator.

5 The Proposed Cultural Immune System

Our second proposed approach uses the previously described Immune System-based algorithm as the population space of a Cultural Algorithm.Pseudo-Code 2 summarizes the general steps of the proposed algorithm.

```
1.Initialize the population of antibodies;
2.Evaluate the Population;
3.Initialize the Belief Space;
gen←0;
4.While the Termination Criterion is not satisfied
  4.1.Clone the Population;
  4.2.Apply the Influenced Hypermutation to the Clones
    selected according to each knowledge source probability;
  4.3.Apply the Quasi-Simplex Local Search;
  4.4.Apply the Repair Procedure;
  4.5.Evaluate the  Clones;
  4.6.Apply the Aging Operator to the Population and the Clones;
  4.7.Apply the $(\mu+\lambda)$−Selection Operator
  4.8.Apply the Acceptance Function;
  4.9.Update the Belief Space (knowledge sources) using the
    best nAccepted antibodies of the Population;
  4.10.Apply the Main Influence Function;
  4.11.gen = gen +1;
End While
```

Pseudo-Code 2: Proposed Cultural Immune System

To the best of our knowledge it is the first time that an Immune System is studied as the population space of a Cultural Algorithm.

The Cultural Immune System can be considered as an extension of the previous algorithm where the Belief Space and the communication protocols are added to improve the performance of the original Immune System. The belief space is used to extract information from the antibodies' population and uses this knowledge as a guide to generate new antibodies during the hypermutation operators through the influence functions, i.e. the hypermutation operators are replaced by the influence functions. The communication protocols dictates which antibodies will be considered during the update of the belief space (acceptance function) and the probability of a knowledge stored in the belief space to influence a hypermutation operator (main influence function).

5.1 Communication Protocols: Acceptance and Influence Functions

In this work a dynamic acceptance function [3][4] is used. This acceptance function is given by:

$$nAccepted = popSize * (accept_{perc} + \frac{accept_{perc}}{gen}) \tag{5}$$

where $nAccepted$ are the number of antibodies of the population that will be used to update the knowledge sources, $popSize$ is the number of antibodies in the population, $accept_{perc}$ is a parameter that determines the percentage of accepted antibodies and gen is the current generation. This acceptance function allows more antibodies to contribute to the update during the beginning of the evolution (when there is little accumulated knowledge) and less antibodies at the end of the evolution (when most of the knowledge have already been acquired).

The main influence function is responsible for choosing (using a Roulette Wheel method [7]) the knowledge source that will influence the hypermutation operators. At the beginning, all the knowledge sources have the same probability to be applied (0.25). During the evolutionary process, the probability of the *ith* knowledge source influence a hypermutation operator is given by Equation 6.

$$probKS_k = 0.1 + 0.6 * (\frac{ant_{gbKS_k}}{popSize}) \tag{6}$$

where $probKS_k$ is the probability of selecting the *kth* knowledge source to influence a hypermutation operator, ant_{gbKS_k} is number of antibodies that were generated by an influence of the *kth* knowledge and $popSize$ is the size of the population. Equation 6 favors the application of knowledge sources that are capable of maintaining their generated antibodies in the population and guarantees that each knowledge source has at least a 10% chance of being applied.

5.2 Knowledge Sources

The following knowledge sources are utilized in this work:

Situational Knowledge. The Situational Knowledge stores the $pbest$ antibodies found during the evolutionary process [7]. These antibodies are used as leaders to influence the hypermutation operators. This influence is similar to the hypermutation operation (Equation 4) but the term $ant_{i,j}$ on the right side is substituted by the term $best_{k,j}$, where $best_{k,j}$ is the *jth* component of the *kth* best antibody stored in the Situational Knowledge, and k is an index randomly selected among the best antibodies.

Normative Knowledge. The Normative Knowledge contains the intervals for the power outputs of the generator units where good solutions have been found, and is used to move the outputs of the new solutions towards those intervals. The intervals of the Normative Knowledge are initialized with the lower and upper bounds of the output of the generator units. The update of the normative knowledge can reduce or expand the intervals stored on it. An expansion takes place when the accepted individuals do not fit in the current interval, while a reduction occurs when all the accepted individuals lie inside the current interval, and the extreme values are associated with individuals with better fuel cost [6]. It is important to note that the limits of the normative knowledge can not violate the operational constraints.

The influence of the Normative Knowledge is as follows:

$$ant_{i,j} = \begin{cases} ant_{i,j} + mult * Cost_i/MinCost_{gen} * Abs(R(0,1)) * (NUp_j - NLo_j), \\ \text{if } ant_{i,j} < NLo_j \\ ant_{i,j} - mult * Cost_i/MinCost_{gen} * Abs(R(0,1)) * (NUp_j - NLo_j), \\ \text{if } ant_{i,j} > NUp_j \\ ant_{i,j} + mult * Cost_i/MinCost_{gen} * R(0,1) * (NUp_j - NLo_j), \\ \text{otherwise} \end{cases} \tag{7}$$

where NUp_j and NLo_j are, respectively, the upper and lower bounds of the normative interval associated with the *jth* component, $Abs(x)$ is a function that returns the absolute value of x and the other terms are as defined in Equation 4. This influence function is adaptive: it is intensive when the normative interval is large (the good interval is very uncertain) and it is smooth when the normative interval is small (the good interval is known).

Historical Knowledge. This knowledge source was introduced in Cultural Algorithms as a mean to adapt to environmental changes [5]. It stores a list of the best antibodies found before the last $HistorySize$ environmental changes. It also stores the average direction and the average distance (size) of the changes for each component between the environmental changes. In this work, since there is no environmental change, this knowledge is adapted and it happens when the algorithm is trapped at a local optimum (there is no change in the best antibody found during the last p generations). The influence function of the Historical Knowledge used in this work is:

$$ant_{i,j} = \begin{cases} ant_{i,j} + mult * Cost_i/MinCost_{gen} * Abs(R(0,1)) * (AvDist_j), \\ \text{if } AvDir_j \geq 0 \\ ant_{i,j} - mult * Cost_i/MinCost_{gen} * Abs(R(0,1)) * (AvDist_j), \\ \text{if } AvDir_j < 0 \end{cases} \tag{8}$$

where $AvDist_j$ is the average distance change in the *jth* component, $AvDir_j$ is the average direction of the change in the *jth* component (both are given by Equation 9) and the other terms are as previously defined.

$$\begin{aligned} AvDist_j &= (\textstyle\sum_{i=1}^{HistorySize-1} Abs(HistoryBest_{i+1,j} - HistoryBest_{i,j}))/ \\ &\quad (HistorySize - 1) \\ AvDir_j &= \ (\textstyle\sum_{i=1}^{HistorySize-1} Sign(HistoryBest_{i+1,j} - HistoryBest_{i,j})) \end{aligned} \tag{9}$$

where $HistorySize$ is the number of antibodies stored in historical knowledge, $HistoryBest_{i,j}$ is the jth component of the ith best antibody and $Sign(x)$ is a function that returns +1 if x is positive, -1 if x is negative and 0 otherwise.

This influence tries to increment the *jth* component of the antibody submitted to hypermutation if in average the *jth* component of new best antibody was greater than or equal to the *jth* component of the previous best antibody and tries to decrement this component otherwise. In either case the hypermutation is proportional to the average distance observed between changes (so if the new best antibodies are found far away from the previous ones the hypermutation will be intensive and it will be smooth if they are found near the previous ones).

Topographical Knowledge. The Topographical Knowledge is used to create a map of the fitness landscape of the problem during the evolutionary process. It consists of a set of regions and the best individual found on each region. It also stores a sorted list of the r best regions (which is sorted according to its best antibody). A region is represented as a node in a binary tree and stores the upper and lower bounds for each component and the best antibody found so far in that region. The binary tree is initialized with a root node that represents the entire feasible space and has the best antibody of the first generation. If during the update of the Topographical Knowledge an antibody with a better fuel cost than the best antibody represented in a region is accepted and this antibody belongs to this region, the node is split (the region is divided in two). The component where the division is made is the component where there is the highest difference between the previous best antibody of this region and the new one. The division is made in half the distance between the value of the split's component of the new and the old best antibodies of the region.

$$SplitPoint = \begin{cases} New_j + (Old_j - New_j)/2, \\ \text{if } Old_j \geq New_j \\ Old_j + (New_j - Old_j)/2, \\ \text{otherwise} \end{cases} \tag{10}$$

where $SplitPoint$ is the value of the jth component where the region will be divided, Old_j is the jth component of the previous best antibody of the cell being divided and New_j is the jth component of the new best antibody found in this cell.

It is important to note that only leaf nodes can be stored in the ordered list or split. The influence function of the Topographical Knowledge is described by Equation 11.

$$ant_{i,j} = ant_{i,j} + mult * Cost_i/MinCost_{gen} * R(0,1) * (R^{upper}_{k,j} - R^{lower}_{k,j}) \tag{11}$$

where R^{upper}_k and R^{lower}_k are the upper and lower bounds for the *jth* component in the *kth* region where is randomly selected according to the affinity of the best antibody of each region if a uniform random number in the interval $[0, 1]$ is less than $pElite$ (probability of the best regions be chosen more often) and randomly selected independently of the affinity otherwise. The other terms of the equation are as previously defined. This influence function tends to explore good regions of the search space.

6 Experiments and Results

Three test cases were used to validate the proposed Immune Systems: 13 generators with a load demand of 1800 MW [22], 13 generators with a load demand of 2520 [24] MW and 40 generators with a load demand of 10500 MW [22].

The algorithms were executed 50 independent times with the following set of parameters (when applicable): $popSize = 50$, termination criterion = 3000 generations, $dup = 4$, $\varphi = 40$, $\rho = 4$, $\tau_B = 100$, $pbest = 10$, $HistorySize = 10$, $p = 200$, $accept_{perc} = 0.2$. Although the great number of parameters, the proposed Immune Systems are little influenced by them. This fact was observed during the experiments realized to find this particular good set of parameters (the adjustment tests were done using only the first test case).

In order to validate our methodologies, we compared the proposed immune-based systems with other state-of-the-art approaches. Although it would be better to compare

Table 1. Results for the Test Cases. Costs are given in US$.

<table>
<tr><th colspan="5">Case with 13 Generators and Load Demand of 1800 MW</th></tr>
<tr><th>Method</th><th>Minimum Cost</th><th>Mean Cost</th><th>Maximum Cost</th><th>Standard Deviation</th></tr>
<tr><td>IS</td><td>17960.91</td><td>17983.03</td><td>18074.41</td><td>24.96</td></tr>
<tr><td>CIS</td><td><b>17960.73</b></td><td><b>17970.35</b></td><td><b>17977.90</b></td><td><b>1.87</b></td></tr>
<tr><td>DEC-SQP [22]</td><td>17963.94</td><td>17973.13</td><td>17984.81</td><td>1.97</td></tr>
<tr><td>IGA [26]</td><td>18063.58</td><td>18096.40</td><td>18293.47</td><td>45.79</td></tr>
<tr><td>MPSO [19]</td><td>17973.34</td><td>-</td><td>-</td><td>-</td></tr>
<tr><th colspan="5">Case with 13 Generators and Load Demand of 2520 MW</th></tr>
<tr><th>Method</th><th>Minimum Cost</th><th>Mean Cost</th><th>Maximum Cost</th><th>Standard Deviation</th></tr>
<tr><td>IS</td><td>24164.80</td><td>24223.23</td><td>24427.29</td><td>52.74</td></tr>
<tr><td>CIS</td><td>24164.79</td><td>24181.67</td><td>24255.84</td><td>30.16</td></tr>
<tr><td>SDE [23]</td><td><b>24164.05</b></td><td><b>24168.28</b></td><td><b>24200.05</b></td><td>-</td></tr>
<tr><td>DTSA [24]</td><td>24169.95</td><td>-</td><td>-</td><td>-</td></tr>
<tr><th colspan="5">Case with 40 Generators and Load Demand of 10500 MW</th></tr>
<tr><th>Method</th><th>Minimum Cost</th><th>Mean Cost</th><th>Maximum Cost</th><th>Standard Deviation</th></tr>
<tr><td>IS</td><td>121529.26</td><td>121880.13</td><td>122283.55</td><td>237.09</td></tr>
<tr><td>CIS</td><td><b>121500.43</b></td><td><b>121735.26</b></td><td><b>122142.74</b></td><td><b>149.02</b></td></tr>
<tr><td>DEC-SQP [22]</td><td>121741.97</td><td>122295.12</td><td>122839.29</td><td>386.18</td></tr>
<tr><td>NPSO-LRS [21]</td><td>121664.43</td><td>122209.31</td><td>122981.59</td><td>-</td></tr>
<tr><td>CEP-PSO [25]</td><td>123670.00</td><td>124145.60</td><td>124900.00</td><td>-</td></tr>
</table>

the algorithm with the same set of algorithms for all test cases, it was impossible due to the fact that the other works were applied to only one or two of the test cases. Table 1 summarizes the results obtained for the three test cases.

For the test case of 13 generators with a load demand of 1800 MW [22]. The table shows that the best results were obtained with the Cultural Immune System (CIS). The Immune System algorithm (IS) found the second best result concerning the minimum cost, outperforming the non-immune algorithms.

Table 1 also presents the comparison of the results obtained by our Immune Systems and other methods reported in the literature for the test case with 13 generators and a load demand of 2520 MW. For this test case the SDE algorithm achieved better results than both Immune Systems. Despite of this, the minimum fuel cost obtained by CIS and IS are less than one dollar more than the minimum fuel cost of the SDE and when compared with DTSA it is more than five dollars lower.

The comparison for the test case with 40 generators and load demand of 10500 MW shows that the proposed CIS was able to attain better values with respect to the statistics than the other algorithms. For this test case the IS was able to outperform all non-immune approaches reported in the literature in all statistics, losing only for CIS.

Based on a ranksum test [27] with 95% degree of confiability it is possible to conclude that the quality of the results obtained by CIS are better than those obtained by IS, attesting the superiority of the cultural method. The p-values obtained were 0.013, $7 * 10^{-8}$ and 0.0041 for the first, second and third test case, respectively. Due to the lack of data given by the other compared methods, we were not able to realize statistical tests comparing these methods and the proposed Cultural Immune System.

7 Conclusions and Future Researches

In this paper we implemented two Immune Systems based on the Clonal Selection Principle. Both algorithms applied static cloning operator, Gaussian and Cauchy Hypermutations, static pure aging, $(\mu + \lambda)$-selection operator and inversely proportional mutation potential. This set of operators proved to be very effective in finding good solutions to the economic load dispatch problem with valve-point effect, which is an important and complex practical engineering problem.

Although the adjustment of the parameters was done on the first test case, the same set of parameters was able to tackle the other problem instances successfully, what corroborates to the robustness of the method.

Our proposed novel approach, the Cultural Immune System (CIS), was capable of improving the performance of its non-cultural counterpart. CIS compared favorably with state-of-the-art algorithms for two of the test cases improving the best reported minimum cost, maximum cost, mean cost and standard deviation of the costs. Although it was outperformed in the second test case by SDE, the minimum cost found by both algorithm are comparable. Because of these facts we assume that CIS can be considered one of the state-of-the-art algorithm for the economic load dispatch problem.

In future works we intend to analyze the behavior of the proposed algorithms in other engineering problems. We also intend to substitute the adaptive multiplier used on the hypermutation operators by a chaotic multiplier that has been shown to improve the performance of a differential evolution algorithm [22] and to incorporate new knowledge sources, especially the probabilistic ones.

References

1. de Castro, L.N.: Fundamentals of Natural Computing: basic concepts, algorithms, and applications. Chapman & Hall/CRC, Sydney, Australia/Boca Raton, USA (2006)
2. Reynolds, R.G., Peng, B., Brewster, J.: Cultural Swarms II: Virtual Algorithm Emergence. In: Proceedings 2003 IEEE Proceedings of Congress on Evolutionary Computation, Canberra, Australia, December 8-12, 2003. IEEE Computer Society Press, Los Alamitos (2003)
3. Iacoban, R., Reynolds, R.G., Brewster, J.: Cultural Swarms: Modeling the Impact of Culture on Social Interaction and Problem Solving. In: Proceedings 2003 IEEE Proceedings of Congress on Evolutionary Computation, Canberra, Australia, December 8-12, 2003, pp. 205–211. IEEE Computer Society Press, Los Alamitos (2003)
4. Reynolds, R.G., Peng, B.: Cultural algorithms: Computational Modeling of How Cultures Learn to Solve Problems: an Engineering Example. Cybernetics and Systems 36(8), 753–771 (2005)
5. Reynolds, R.G., Peng, B.: Knowledge Integration On-The-Fly in Swarm Intelligent Systems. ICTAI, 197–210 (2006)
6. Becerra, R.L., Coello, C.A.C.: Optimization with Constraints using a Cultured Differential Evolution Approach. In: Beyer, H.-G., et al. (eds.) Genetic and Evolutionary Computation Conference (GECCO'2005), June 2005, vol. 1, pp. 27–34. ACM Press, Washington, DC, USA (2005)

7. Becerra, R.L., Coello, C.A.C.: Cultured Differential Evolution for Constrained Optimization. Computer Methods in Applied Mechanics and Engineering 195(33-36), 4303–4322 (2006)
8. Saleem, S.M.: Knowledge-Based Solution to Dynamic Optimization Problems using Cultural Algorithms. PhD thesis, Wayne State University, Detroit, Michigan (2001)
9. Cortés, N.C., Trejo-Pérez, D., Coello, C.A.C.: Handling Constraints in Global Optimization using an Artificial Immune System. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 234–247. Springer, Heidelberg (2005)
10. de Castro, L.N., Von Zuben, F.J.: Learning and Optimization Using the Clonal Selection Principle. The Special Issue on Artificial Immune Systems of the journal IEEE Transactions on Evolutionary Computation 6(3) (June 2002)
11. Cutello, V., Nicosia, G., Pavone, M.: Exploring the Capability of Immune Algorithms: A Characterization of Hypermutation Operators. In: Third International Conference on Artificial Immune Systems, pp. 263–276 (September 2004)
12. 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)
13. Cutello, V., Narzisi, G., Nicosia, G., Pavone, M.: ‘Real Coded Clonal Selection Algorithm for Global Numerical Optimization using a new Inversely Proportional Hypermutation Operator. In: The 21st Annual ACM Symposium on Applied Computing, SAC 2006, Dijon, France, April 23-27, 2006, vol. 2, pp. 950–954. ACM Press, New York (2006)
14. Cutello, V., Nicosia, G., Oliveto, P.S., Romeo, M.: On the Convergence of Immune Algorithms. In: The First IEEE Symposium on Foundations of Computational Intelligence, FOCI 2007. IEEE Computer Society Press, Los Alamitos (2007)
15. Almeida, C.P., Gonçalves, R.A., Delgado, M.R.B.: A Hybrid Immune-Based System for the Protein Folding Problem. LNCS. Springer, Heidelberg (2007)
16. Wood, A.J., Wollenberg, B.F.: Power Generation, Operation, and Control. John Wiley & Sons, Inc. New York (1984)
17. Zhang, G., Lu, H.Y., Li, G., Xie, H.: A New Hybrid Real-Coded Genetic Algorithm and Application in Dynamic Economic Dispatch. In: Proceedings of the 6th World Congress on Intelligent Control and Automation, June 21-23, 2006 (2006)
18. Liu, H., Ma, Z., Liu, S., Lan, H.: A New Solution to Economic Emission Load Dispatch Using Immune Genetic Algorithm. In: IEEE Conference on Cybernetics and Intelligent Systems. IEEE Computer Society Press, Los Alamitos (2006)
19. Hou, Y., Lu, L., Xiong, X., Wu, Y.: Economic Dispatch of Power Systems Based on the Modified Particle Swarm Optimization Algorithm. In: IEEE/PES Transmission and Distribution Conference and Exhibition. IEEE Computer Society Press, Los Alamitos (2005)
20. Park, J., Jeong, Y., Lee, W.: An improved particle swarm optimization for economic dispatch problems with non-smooth cost functions. IEEE Power Engineering Society General Meeting (2006)
21. Immanuel Selvakumar, A., Thanushkodi, K.: A New Particle Swarm Optimization Solution to Nonconvex Economic Dispatch Problems. IEEE Transactions on Power Systems 22(1), 42–51 (2007)
22. Coelho, L.S., Mariani, V.C.: Combining of Chaotic Differential Evolution and Quadratic Programming for Economic Dispatch Optimization With Valve-Point Effect. IEEE Transactions on Power Systems 21(2) (2006)
23. Balamurugan, R., Subramanian, S.: Self-Adaptive Differential Evolution Based Power Economic Dispatch of Generators with Valve-Point Effects and Multiple Fuel Options. International Journal of Computer Science and Engineering 1(1) (2007)

24. Khamsawang, S.P., Boonseng, S.: Distributed tabu search algorithm for solving the economic dispatch problem. In: TENCON 2004, vol. C, pp. 484–487 (2004)
25. Sinha, N., Purkayastha, B.: PSO embedded evolutionary programming technique for non-convex economic load dispatch. IEEE PES - Power Systems Conference and Exposition 1, 66–71 (2004)
26. Ling, S.H., Lam, H.K., Leung, F.H.F., Lee, Y.S.: Improved genetic algorithm for economic load dispatch with valve-point loadings. In: The 29th Annual Conference of the IEEE Industrial Electronics Society - IECON ’03, vol. 1, pp. 442–447. IEEE Computer Society Press, Los Alamitos (2003)
27. Conover, W.J.: Practical Nonparametric Statistics. Wiley, New York (1980)

Artificial Immune System to Find a Set of *k*-Spanning Trees with Low Costs and Distinct Topologies

Priscila C. Berbert, Leonardo J.R. Freitas Filho, Tiago A. Almeida,
Márcia B. Carvalho, and Akebo Yamakami

Department of Telematics, School of Electrical and Computer Engineering,
State University of Campinas, Campinas, São Paulo, PO Box 6101, Brazil
{pberbert,lfreitas,tiago,marciab,akebo}@dt.fee.unicamp.br

Abstract. This work proposes an Artificial Immune System to find a set of *k* spanning trees with low costs and distinct topologies. The attainment of this set of solutions is necessary when the problem has restrictions or when the interest is to present good alternative solutions for posterior decision making. Solving this problem means to explore an enormous space of solutions that grows as the number of graph nodes increases; it becomes impractible using exact or comparative methods. However, it is known that bio-inspired algorithms have a high capacity of exploration and exploitation of the search space. Moreover, inherent characteristics of AIS become the search mechanism more efficient allowing the resolution of this problem in a feasible computational time.

Keywords: Bio-inspired Computing, Artificial Immune Systems, Spanning Trees.

1 Introduction

A large number of engineering applications uses graph theory nowadays. This theory is used in the most distinct areas as telecommunications, logistics, manufacturing, etc. Minimum Spanning Tree (MST) problem shows up in many applications. One example is to lay phone or electric lines between a set of localities using the highways infrastructure while minimizing material costs. There are efficient algorithms to solve this kind of problem, as *Kruskal*, *Prim* and *Sollin* [1].

A large number of graph problems, even with finite number of possibilities, can not be solved in a feasible computational time [1]. One reason is that these kind of problems become more complex as the number of nodes increases.

When the problem has restrictions or it is necessary to offer good alternative solutions for posterior decision-making, then a set of Spanning Trees (ST) with low costs and distinct topologies should be found out. One difficulty is that the number of possible solutions can be huge, and then the attainment of the solution set composed by ST with low costs through precise methods is impracticable.

Bio-inspired algorithms, like Artificial Immune Systems (AIS), Genetic Algorithms, Evolutive Strategies, Ant Colony Optimization, etc., are attractive methods frequently used to solve a large number of optimization problems [6]. The great potential of these algorithms is the integration of the ample search space exploration

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 395–406, 2007.

with a process of located search called exploitation. This integration results in a high robustness level that allows the application of the bio-inspired computation in many practical problems, which other strategies are inefficient [6].

According to Dasgupta [5], AIS are computational mechanisms composed by intelligent methodologies, inspired in the biological immune system, to solve problems of the real world; many of its properties are advantageous in diverse applications.

In this paper, it is proposed a bio-inspired algorithm based on AIS to find a solution set composed by the *k*-Spanning Trees with low costs and distinct topologies. The remainder of this paper is organized as follows. Section 2 introduces the problem. In Section 3, related works are presented. In Section 4, concepts of biological immune system are introduced. Section 5 proposes an implementation to attain the solution set. In Section 6, experiments results are presented. Finally, Section 7 offers the conclusions.

2 Problem Presentation

Given a graph *G*, our aim is to find a set of *k*-spanning trees ($k \in \mathbb{N}^+$ and $k > 1$), with distinct topologies and low costs associated. Among many possible applications, the result can be used in problems with restrictions or when it is necessary to present a set of good solutions for a posterior decision-making. Due to the combinatorial nature of the problem, where the number of possible solutions grows as increases the number of nodes, it is necessary to use an efficient boarding capable to explore the search space of solutions in reasonable computational time.

The number of spanning trees in a complete graph, *G: (N, A)*, where *N* is the set of nodes and *A* is the set of the edges, is in the order of m^{m-2}, $m = |N|$ [13]. For instance, a complete graph with 50 nodes spans about 50^{48} possible trees. So, it is infeasible to perform an exhaustive search, because assuming that 1000 computers analyze 1000 trees per second, each one, it would take approximately 10^{68} years to evaluate all possible solutions (it is estimated that the universe has about 10^{13} years) [3]. However, AIS based on techniques of clonal selection is considered aiming to find a good set of solutions as well as avoiding the complexity issue. Moreover, characteristics inherit from immune system, as diversity maintenance, memory, mutation and proliferation proportional to the affinity are important for the attainment of good results [6].

3 Related Works

Some algorithms had been proposed to find all spanning trees of a graph, as Minty [10]. However, none of them consider the edge costs for constructing the trees. So, it is necessary to classify all possible solutions in an increasing cost sequence, that is, there is the necessity of compare all spanning trees with each one; but it is computational impracticable. Sörensen and Janssens [14], inspired in Murty [11], considered an algorithm to find all spanning trees in increasing cost sequence. The algorithm is based on partitions sets, where each partition is defined as $p = \{a_1,...,a_r;\overline{b_1},...,\overline{b_s}\}$, that is, the set of all spanning trees contend all edges a_1, ..., a_r and none of edges of b_1,

..., b_s. It is initiated with the MST and the partitions set $\{p_1, p_2, \ldots, p_{m-1}\}$; in period *i* the *i*-st spanning tree is found. Because of the *bottom-up,* explore character, from the neighborhood of MST and expanding to all the search space, the algorithm becomes inefficient for instances that the *k* best spanning trees are spread in the search space. Another difficulty could be found when it would be necessary to obtain a large number of spanning trees for a graph with many nodes.

Almeida and Yamakami [3] had considered Evolutive Algorithms to solve the problem of minimum spanning tree with *fuzzy* parameters. In this work, graphs with known structure, but with uncertainties associated to the edge costs had been studied. The objective of the algorithms is to find the solution set composed by trees that have high possibility degree of being the best solution.

4 Biological Immune System

This section brings an introduction to basic concepts of biological immune system; the presented approach had been inspired in De Castro and Von Zuben [6].

The natural immune system is complex and possesses a large variety of components and functionalities. To develop a computational system inspired in the natural one, a simplification of the reality becomes necessary to fit the biological concepts, considered more important, for application in the proposed problem.

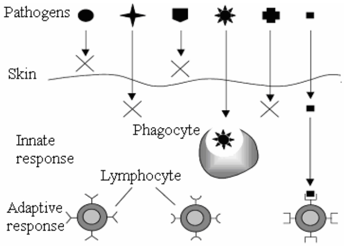


Fig. 1. Multi-layer structure of the Immune System [6]

Conceptually, Immune System (IS) is a collection of molecules, cells and organisms whose complex interaction in an efficient system is usually capable to protect an individual of invaders. The IS is subdivided in Innate Immune System, formed by phagocytes cells responsible for a first combat to the pathogens, and Adaptive Immune System, formed for cells (lymphocytes) that evolve to provide more versatile defenses ways and a bigger level of protection face to new infections for the same agent (Figure 1). The characteristics that foment the AIS theory are based on the functionalities of the Adaptive Immune System. The lymphocytes are submitted to a process similar to natural selection [8], where only cells that found an antigen with which receptors can interact, will be activated to proliferate generating clones. These clones will differentiate in plasma cells, that secrete antibodies with the same receptors characteristics; or memory cells, when the immune system will be exposed again to one determined antigen, these cells are activated in order to get a more efficient future reply.

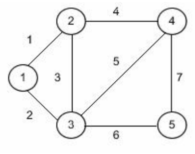
On the processes of proliferation and differentiation of the high affinity cells, the somatic hypermutation is realized to refine, through exchanges introduced into the variable region genes, the immune reply to the recognized antigen. This mechanism allows the creation of antibodies capable to recognize the antigen with more efficiency. Generation and mutation mechanisms provide a powerful antigen recognition, because the immune system can produce a practically infinite number of cellular receptors from a finite genome. The antibodies perform the main role in Immune System. They are capable to adhere to the antigen, to neutralize them and to mark them for that other immune system cells eliminate them [8].

5 The Proposed Artificial Immune System

AIS inspired in the concepts presented in Section 4 is proposed. The repertoire is formed by antibodies (spanning trees) that generate new heirs (clones) with the same characteristics of father-antibody. These clones evolve by means of maturation operators. An affinity measure is used to privilege the proliferation of the more adapted antibodies (clonal selection), that is, only those antibodies that find an antigen with which receptor can interact will be activated to proliferate and to differentiate.

5.1 Representation

Let a graph G: (N, A), being $m = |N|$, a spanning tree have $m - 1$ edges that connect all nodes of G without cycles [1]. Due to this characteristic, it was proposed a representation that makes possible the treatment of each antibody as being a spanning tree, that is, only feasible solutions are considered. Given the graph of Figure 2 and its respective assignment matrix MD^{wx3}, w = number of graph edges:



Index	Node i	Node j	Cost
1	1	2	Cost 1
2	1	3	Cost 2
…	…	…	…
7	4	5	Cost 7

Fig. 2. A graph and its respective assignment matrix

Each antibody has m -1 positions that can contain elements of 1 to w, where each position is associated to an edge that is part of the tree in question.

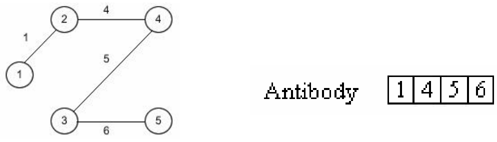


Fig. 3. Codification example

Figure 3 presents an example of codification of the spanning tree in an antibody. The repertoire of antibodies is formed by *n* (user defined parameter) spanning trees; ambiguity (different antibodies representing the same ST) and infeasible solutions are not allowed.

5.2 The Initial Repertoire

The initial repertoire is randomly generated, covering the search space with antibodies representing feasible solutions. For this procedure, random costs are attributed to graph edges and *Kruskal* algorithm is executed. The process is repeated to complete the repertoire.

5.3 Affinity Measure

Let *T** the MST obtained by *Kruskal* algorithm. The measure that evaluates the quality of antibody T_i in relation to others of the repertoire is defined by:

$$Affinity\ (T_i) = \frac{cost\ (T^*)}{cost\ (T_i)}\ . \tag{1}$$

The $Affinity\ (T_i) \in [0,1]$ and as closer are the costs T_i and T^* as greater will be its affinity.

5.4 Proliferation

Antibodies with $Affinity(T_i) \geq \lambda$ are selected for proliferation, considering the affinity threshold $\lambda \in (0,1)$. Then, $n_c(T_i)$ clones, identical to selected father-antibody T_i, are generated (Figure 4). The number of clones produced by each antibody is calculated by the Equation (2).

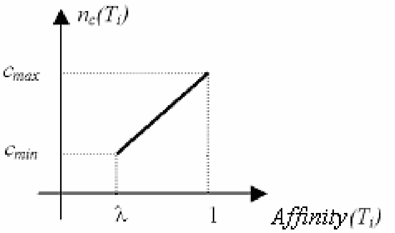


Fig. 4. Number of clones for each antibody in function to its affinity

Assuming that, $n_c(T_i) \in [c_{\min}, c_{\max}]$; $c_{\min}$, $c_{\max}$ and $n_c(T_i) \in$ N.

$$n_c\left(T_i\right) = \left\lfloor c_{\max} + \left(\frac{c_{\max} - c_{\min}}{1-\lambda}\right)\left(Affinity\ \left(T_i\right) - 1\right)\right\rfloor\ . \tag{2}$$

5.5 Maturation

Maturation process is based on Almeida and Yamakami [3] proposal. The immune-inspired concept of proportional inversely mutation to the affinity is also considered. The repertoire of clones *C* is submitted to the affinity maturation process (mutation) generating a new repertoire *C'*. This mutation occurs choosing an edge that composes the clone and changing it for another one that belongs to cut set, as shown in the follow sequence (Figure 5):

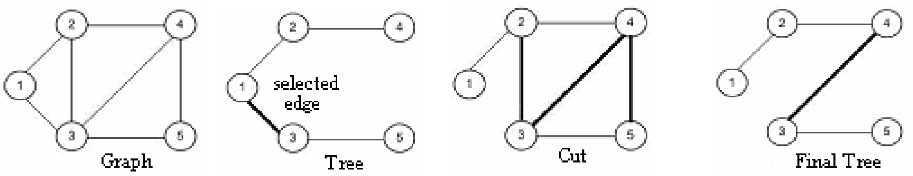


Fig. 5. The process of maturation with one edge exchange, 1-*opt*

The mutation depends on the father-antibody affinity and varies from 1-*opt* until β-*opt* (Figure 6).

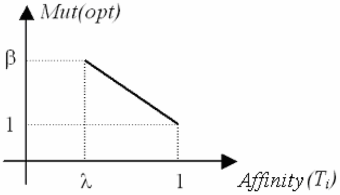


Fig. 6. Mutation realized in the clones

What characterizes the selected mutation is the amount of edges changed in the tree (clone). This number is calculated based on the graph of Figure 6:

$$Mut(opt) = \left\lfloor 1 + \left(\frac{1-\beta}{1-\lambda}\right)(Affinity(T_i) - 1) \right\rfloor, \tag{3}$$

$\beta \in \mathrm{N}$ is the maximum number of edges exchanges.

The number of clones generated by the antibody and the number of mutations realized in the clones depends on the affinity measure. While the number of mutations is inversely proportional to antibody affinity, the capacity for generating clones is directly proportional. Antibodies with high affinity generate more clones and they are submitted to few mutations, because the interest is to *intensify* the local search. Note that, this characteristic improves the *intensity* of local search. On the other hand, antibodies with low affinity are submitted to many mutations and they generate few clones. Therefore, the high number of mutations makes possible a great *diversification* of the solutions, promoting a better exploration of the search space. This is an inherent aspect to the immune-based approach.

5.6 Selection

The selection procedure of clones is inspired on principles of biological Clonal Selection [6]. The choose method is elitist [9], thus, the clone with highest level of affinity is privileged. The most adapted clone (with smallest cost) generated by each antibody is selected and compared with its father-antibody; only the best one is preserved in the repertoire.

5.7 Diversity

The maintenance of high taxes of diversity between the candidate solutions is a strong characteristic of the natural immune system [6]. Besides, the performance of the proposed algorithm is directly related with the existence of mechanisms that control and keep the diversity between the antibodies. A random initial repertoire attends in part this requirement. For diversity control during the algorithm iterations, it was proposed a metric based on the number of exchanges among edges necessary to transform a spanning tree into another one (see graph example *G: (N, A)*, Figure 7).

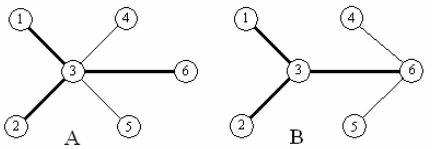


Fig. 7. Visualization of edges sharing between two trees

Trees A and B above share between itself three edges; from A to B, it is necessary to change {(3,4), (3,5)} for {(4,6), (5,6)}, and vice-versa. So, to evaluate the distance between two solutions A and B, it was defined a metric given by the expression (4):

$$D(A,B) = (m-1) - (number\ of\ \ shared\ edges). \tag{4}$$

For trees A and B given by Figure 7, $D(A,B) = (6\ -\ 1)$ - 3 = 2. Note that, the maximum distance between the trees is D_{max} = m - 1. Using this equation is possible to identify when the repertoire is losing diversity, that is, when the trees have similar topologies. This metric gives subsidy to propose a *suppression* mechanism to avoid that many antibodies concentrate only in few promising regions.

An example for repertoire evolution is observed in the Figure 8. Initially, the antibodies are spread in the search surface. After some iterations, the tendency is that the solutions will concentrate on regions of local optimum. However, the presence of many antibodies in these regions is not necessary because the capacity of exploitation already is high due to the proliferation mechanism. To identify antibodies to be suppressed, a minimum distance that they must have between themselves is calculated. Thus, only one potential antibody to represent a determinate region is kept and all others are substituted.

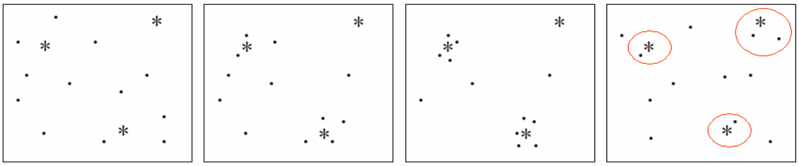


Fig. 8. Illustration of the suppression mechanism, * *k*-best spanning trees, • Antibodies

However, computational cost to verify the repertoire diversity is high, since each antibody must be compared with all other ones. Hence, *suppression* should be applied periodically on resultant antibodies of proliferation; in remaining antibodies, this procedure is not necessary because they will be substituted by other ones randomly generated. Moreover, the suppression allows that new regions can be explored from the insertion of these randomly generated antibodies.

Other considered mechanism is inspired on *Natural Selection*, because antibodies with low affinity, which never were selected for proliferation, should be replaced for other ones. This process is in accordance with the concept of *Available Repertoire*, because antibodies were not activated until the moment, but can be used [6]. The procedure of antibodies substitution can be understood as an exploration in the search space; and the principle of clonal selection provides a local search in different directions, the exploitation [4].

5.8 Memory

Memory is necessary when a load of current information is available but it is not interesting to carry it along the iterations. However, the information should not be lost because its content could be important in the future; in this situation, it must be stored. When the initial repertoire is generated, the *T** got through *Kruskal* and the *k*-1 better antibodies are placed on the memory (elitism). At the end of each iteration, the memory, that already contains *k* antibodies, receives another group with other *k* ones. Then a mechanism of memory update is applied reducing it again to *k* distinct antibodies. In this way, when a promising father is substituted by a better son, the father still has the possibility to remain in the external memory, preventing the loss of information.

5.9 AIS Pseudocode

```
T*  ← Find the minimum spanning tree of G
T   ← Initialize the repertoire with n antibodies (trees)
f   ← Calculate the affinity for repertoire T
generation ← 1
while stopped criterion not reached do
   for each antibody $T_i$ do
      if affinity($T_i$) ≥ λ then
         Select the antibody $T_i$ to proliferate
         Calculate $n_c$($T_i$)
         C ← $n_c$($T_i$) clones of $T_i$
         Calculate Mut(opt)
         C′ ← Maturation of clones C, in accordance with Mut(opt)
         $f_c$ ← Calculate the affinity for the clones C′
```

```
            c* ← Select the best clone of C′
            if affinity(c*) ≥ affinity($T_i$) then
               $T_i$ ← c*
            end - if
         end - if
      end - for
      *Verify diversity (Suppression)
      Update the memory
      Substitute not selected antibodies $T_i$
      Update f
      generation ←  generation + 1
   end - while
```

6 Computational Experiments

The AIS was implemented in *MATLAB*® 7.0, and all tests was executed in a platform *Celeron*® *M*, *1.5Ghz* and *512Mb RAM*. Through tests realized with randomly generated initial repertoire, for the instances of TSPLIB[1], Okada [12] and Ali *et al.* [2], it was observed that the *k*-best spanning trees found have many shared edges with *T**, that is, they were in its neighborhood. Therefore, the vaccines concept was used, as proposed by Keko *et al.* [7], in order to improve the algorithm performance. Some considerations about the presented general algorithm in the Section 5 are described below.

The biological immune system is also susceptible to other influences as vaccines (artificial way to raise the individual immunity). In AIS, the vaccines can be viewed as abstract forms of priori knowledge about the problem. Thus, the initial repertoire was created around *T** in the following way: some edges of *T** are randomly selected and they are replaced by other ones in order to construct new spanning trees. Since the repertoire always are restricted to the region around *T**, the antibodies have many shared edges with it. So, the diversity control must take into consideration this characteristic, since an antibody is considered distinct if it possess at least one edge different of other ones. The suppression was applied only on accurately equal antibodies, removing them from the repertoire and substituting them for other trees generated from *T**. This same process was used to replace antibodies that never were selected to proliferate.

The algorithm parameters were adjusted from these considerations and the values presented below referring the best test run. In the total, 30 test runs had been performed, with variation of: iterations number, repertoire size, λ, $c_{\min}$, $c_{\max}$ and β.

$\beta = 1$: the repertoire and the *k*-best solutions are restricted to the neighborhood of *T**, so it is not interesting a high *Mut(opt)*.

$c_{\min} = 8$ and $c_{\max} = 10$: in order to improve the exploration, the number of clones should be high and the mutation probability should be low.

$\lambda = 0.85$: the repertoire affinity is high because the antibodies are around *T**.

Moreover, it was considered *k* =10, a repertoire with 50 antibodies and a maximum number of 60 iterations. The *Instances*, the number of *Nodes*, the cost of

[1] http://www.iwr.uni-heidelberg.de/groups/comopt/soft/TSPLIB95/TSPLIB.html

Table 1. Solution found, in the best test run

Inst.	*C(T*)*	*Costs of k-best trees found*	*t(s)*
Bayg29	1319	[1319, 1320, 1323, 1323, 1323, 1324, 1324, 1324, 1324, 1324]	11.9
Bays29	1498	[1498, 1498, 1498, 1499, 1500, 1502, 1505, 1505, 1506, 1506]	8.2
Brazil58	17514	[17514, 17516, 17518, 17531, 17541, 17554, 17558, 17561, 17568, 17574]	75.1
Si175	20762	[20762, 20762, 20762, 20762, 20768, 20773, 20774, 20776, 20779, 20783]	114.3
Swiss42	1079	[1079, 1079, 1079, 1079, 1079, 1081, 1082, 1082, 1084, 1084]	14.8
Italy 21	2665	[2665, 2665, 2670, 2670, 2670, 2670, 2670, 2675, 2675, 2680]	24.0
USA 70	16743	[16743, 16744, 16749, 16755, 16756, 16759, 16760, 16761, 16761, 16765]	53.8

the MST (T^*), the group $K = \{1,\ldots, k\}$ of solutions and the execution time are presented in Table 1.

It is important to stand out that the instances presented in Table 1 were also tested with randomly generated initial repertoire and the obtained results were similar. However, they spent more time of execution. It leads to the conclusion that, for these instances, the k-best trees are in a region around T^*, so it justifies the use of the vaccines concept.

6.1 New Application Scene

For all instances mentioned in Table 1, the k-best solutions have many shared edges with T^*, thus, it was created a new instance to verify the algorithm exploration potential. A complete graph composed by 29 nodes (*fb29*) which have two minimum spanning trees that share only one edge (Figure 9).

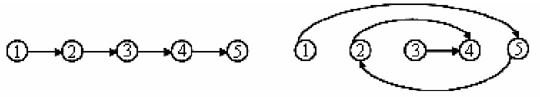


Fig. 9. Construction scheme for *fb29* and an example for an instance with 5 nodes

For the edges in common of these two trees, which were used as base for the instance construction, it was attributed cost 1 and the remains received random costs between 5 and 10. It is important to point out that there are a lot of spanning trees with minimum cost equal to 28, two induced by the construction and others that share edges with low cost.

Again, 30 test runs had been performed, with variation of: iterations number, repertoire size, λ, $c_{\min}$, $c_{\max}$ and β. The initial repertoire and the diversity control were used as defined in Section 5. The parameters of the best test run:

$\beta = 5$, $c_{\min} = 3$, $c_{\max} = 7$.

$\lambda = 0.17$: repertoire affinity is not high because the antibodies are well distributed in the search space, so the affinity threshold must be low. Many antibodies with high affinity arise with the repertoire evolution. To avoid that many antibodies are selected to proliferate, a maximum limit of 35 antibodies was established.

Table 2. Solution found for fb29, in the best test run

Inst.	*C(T*)*	*Costs of k-best trees*	*T(s)*
fb29	28	[28, 28, 28, 28, 28, 28, 28, 28, 28, 28]	80.09

It was considered k = 10, a repertoire with 100 antibodies and a maximum number of 250 iterations. The best solution found are showed in the Table 2.

The instance *fb29* also was tested with the vaccines concept, with the same parameters used for the instances of Table 1; in this in case, the k found solutions have cost 28 and the computational spent time was 17s. However, in this case, the diversity between all found solutions is very low because they were concentrated around the T*. On the other hand, when it was used random initial repertoire, the found solutions present good diversity and they shared a low number of edges. Table 3 shows the distance D, defined in Section 5.7, between each pair of solutions when random initial repertoire was used and when the vaccines concept was used, respectively.

Table 3. Distance between the solutions, for two cases (random initialization, vaccines)

	k_2	K_3	k_4	k_5	k_6	k_7	k_8	K_9	k_{10}
k_1	(16, 1)	(17, 1)	(17, 1)	(11, 1)	(14, 1)	(14, 1)	(14, 1)	(15, 1)	(14, 1)
k_2		(6, 1)	(6, 1)	(14, 2)	(5, 2)	(9, 2)	(6, 2)	(4, 2)	(14, 2)
k_3			(6, 2)	(15, 2)	(7, 1)	(9, 2)	(7, 2)	(8, 2)	(11, 2)
k_4				(15, 2)	(6, 1)	(12, 2)	(8, 2)	(8, 2)	(14, 2)
k_5					(15, 1)	(16, 1)	(14, 1)	(15, 2)	(13, 2)
k_6						(13, 1)	(8, 2)	(7, 2)	(13, 2)
k_7							(7, 2)	(8, 2)	(14, 1)
k_8								(3, 2)	(11, 2)
k_9									(12, 1)

For example, the number of distinct edges between solutions k_1 (first line of Table 3) and k_2 (first column of Table 3) was 16 with the random initialization and only 1 with the use of vaccines concept. It is evident that the random initial repertoire provides more diversity; in this case, the medium distance between the solutions was 10.91. On the other hand, when the vaccines concept was used, the average was 1.57.

6 Conclusions

The proposed algorithm was efficient and at the same time promising for solving the presented problem. Due to the inherent characteristics of the Artificial Immune System, it becomes in a very attractive approach once it have great mechanisms for exploration and exploitation.

Repertoire initialization depends on the kind of desired reply. Random initialization guarantees more diversity between the solutions, but demand more computational time for that good solutions can be found. On the other hand, vaccines concept increases the algorithm performance, but it is necessary to take care with the diversity, because the repertoire already is initialized with similar antibodies and they remain restricted to a region during all the iterations. Moreover, the instance characteristics also can be considered, if the best solutions are concentrated in a region around T^* or if they are spread in the search space.

Mentioned techniques as clonal selection, suppression, number of clones and number of mutations proportional to the affinity influenced a lot to increase the algorithm performance.

The most recent related works are Sörensen and Janssens [14] and Almeida and Yamakami [3]. It was not possible to establish a comparison between our results and the first mentioned reference, because the paper is a theoretical work and the authors did not present computational experiments. In relation to the second reference, it was observed that there is similarity between the modal values of the *fuzzy* costs and the results presented in Table 2.

References

1. Ahuja, R.K., Magnati, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms and Applications, 1st edn. Prentice-Hall, United States (1993)
2. Ali, M., Ramamurthy, B., Deogun, J.S.: Routing and Wavelength Assignment with Power Considerations in Optical Networks. Computer Networks 32, 539–555 (2000)
3. Almeida, T.A., Yamakami, A.: Evolutionary Computation Applied to Solve the Minimum Spanning Tree Problem with Fuzzy Parameters. Master Thesis, School of Electrical and Computing Engineering, State University of Campinas 2006 (in Portuguese)
4. Coello Coello, C.A., Cortés, N.C.: Solving Multiobjective Optimization Problems Using an Artificial Immune System. Genetic Programming and Evolvable Machines 6, 163–190 (2005)
5. Dasgupta, D.: Artificial Immune Systems and Their Applications, 1st edn. Springer, Nova York (1998)
6. De Castro, L.N., Von Zuben, F.J.: Immune Engineering: Development and Application of Computational Tools Inspired in Artificial Immune Systems. PhD Thesis, State University of Campinas, School of Electrical and Computing Engineering 2001 (in Portuguese)
7. Keko, H., Skok, M., Skrlec, D.: Artificial Immune Systems in Solving Routing Problems. Computer as Tool. In: EUROCON 2003. Computer as Tool, The IEEE Region, vol. 8(1), pp. 62–66 (2003)
8. Lederberg, J.: Ontogeny of the Clonal Selection Theory of Antibody Formation. Annals of the New York Ac. of Sc. 546, 175–182 (1988)
9. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs, 3rd edn. Springer, Heidelberg (1996)
10. Minty, G.J.: A Simple Algorithm for Listing All the Trees of a Graph. IEEE Transactions on Circuit Theory CT-12, 120 (1965)
11. Murty, K.G.: An Algorithm for Ranking All the Assignments in Order of Increasing Cost. Operations Research 16, 682–687 (1986)
12. Okada, S.: Interactions Among Paths in Fuzzy Shortest Path Problems. In: Proceedings of the 9th International Fuzzy Systems Associations World Congress, pp. 41–46 (2001)
13. Raidl, R.G., Julstrom, B.A.: Edge-Sets: An Effective Evolutionary Coding of Spanning Trees. Research Report, Vienna University of Technology, Institute of Computer Graphics and Algorithms (2001)
14. Sörensen, K., Janssens, G.K.: An Algorithm to Generate All Spanning Trees of a Graph in Order of Increasing Cost. Operational Research 25, 219–229 (2005)

How to Obtain Appropriate Executive Decisions Using Artificial Immunologic Systems

Bernardo Caldas, Marcelo Pita, and Fernando Buarque

Department of Computing Systems
Polytechnic School of Engineering – University of Pernambuco
Rua Benfica, 455 – Benfica, 50.720-001 – Recife, Pernambuco, Brazil
{bjbc,mrsp,fbln}@dsc.upe.br

Abstract. This paper presents a new selection algorithm based on artificial immunologic systems (AIS) and decision theory. The objective of the algorithm it to increase the performance of executive decisions by identifying more carefully what can be considered an appropriate executive decision to a given specific context. The main idea is to mimic the immunologic system, specifically the self & non-self detection mechanism. According to our proposition, 'decision cells' (analogous to immune-cells) are responsible for selection of the appropriate executive decisions. In this paper we present the motivation, theoretical approach (*i.e.* the analogy between biological and business models), the proposed algorithm and some simulated experiments aiming at real situations.

1 Introduction

The human being organism is a complex system and his functions depend greatly on the effective participation of several subsystems that ought to work together, in a collaborative and harmonious way; this is believed to sustain "life". Among the subsystems that comprise this ingenious machine of well-balanced functionalities, we observed deeply the human immunologic system. The human immunologic system is mainly specialized in defending our organism against infections cased by pathogenic agents (*e.g.*, bacteria, viruses, fungi, etc.), [1], [2]. Further investigations on the mechanisms that govern essential actions of the human immunologic system constitute a challenging theme of research as well as a source of some new approaches for solving real problems; here, the immunologic system was the major inspiration for this work.

Similarly to what was commented above, the principle that governs enterprise systems is quick response time. That is, reactions at the right time and in the proper manner. But this can only happen in the presence of good information, *i.e.* quantity as well as quality. In competitive situations, expedite decision may represent survival or death of the organization, [3], [4], [5], [6]. Sometimes, actually, most of times, the high-volume of available information turns out to be very expensive for the decision process of companies. This happens because it is time consuming to discard all unnecessary information for a given decision task. The current trend is for that problem to get worse as Management Information Systems (MIS) are ubiquitous, and

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 407–419, 2007.

increasing in number and functionalities. Meanwhile, executive decisions remain deprived of selective intelligent support tools.

Some properties of the human immunologic system, such as the capacity of pathogen memorization, the ability to distinguish self and non-self elements, decentralization of actions and learning, [1], [2], were seminal for us to suggest their relevance to executive information systems. Actually, in spite of the short history of Artificial Immunologic Systems (AIS), they are already very useful for computer applications in real world problems, [2], and have an increasing number of published papers on practical applications, [1], [2], [7].

In brief, this work approach is to consider one executive decision as a homeostatic response within a given organization, when information is available. This means that 'decision cell' select appropriate executive decisions when the necessary 'pieces of information' bind appropriately to their 'receptors'. In the human immunologic system, the same happens when antibodies binds to a pathogens.

This paper is organized as follows. Section 2 describes briefly the theoretical concepts that were used in this work, namely, the human immunologic system, decision theory, information and decision systems. Section 3 presents the considered analogy between the human immunologic system with decision systems and the proposed model for selecting of appropriate executive decisions. Sections 4 and 5 describe experiments, results and conclusions.

2 Background

2.1 The Human Immunologic System

The defense mechanism of human (and other animal) organisms against pathogenic agents is carried out by the innate immunologic system. Its functioning is continuously improved by adaptive learning. This adaptive defense mechanism is capable of memorizing specific characteristics of pathogenic agents, acting preventively in future attacks. Working in an intensive and collaborative way, the above mentioned mechanism provides immunity against a great deal of infections.

The non-trivial task of keeping the body secure is made possible by the immunologic response of an efficient layered architecture. The innate immunologic mechanism is mainly composed of two processes called: (*i*) phagocytosis and (*ii*) adaptive immunologic mechanism. Both are mediated by specialized cells. The former, engulf non-self particle and cells, and the latter recognize and destroy antigens, [1], [2], [7], [8], [9], [10], [11], [12], [13].

2.2 Information and Decision

Searching for more suitable ways of decision making is as old as the pre-historic period, when man conducted his decisions by analyzing dreams, animal viscera, smoke, among other elements, [14]. This necessity has been increasingly elaborated through time. Currently, advances in information technology (IT) are of great support for supplying those needs, especially in overcoming human limitations, [15], [16], [17], [18]. Namely, IT may help on not having emotional decisions and help on overcoming the human brain shortcoming of not be able of dealing with too many alternatives (*i.e.* solution scenarios), [19].

Decision making presents two following dimensions: (*i*) the process dimension, which focus is on decision performance (*i.e.* efficiency) and (*ii*) the problem dimension, which focus is on decision efficacy, [3]. Problem is the core of the decision theory. It can be presented in a structured or unstructured manner, the latter characterized by unpredictability, uncertainness, need for innovation, non-uniqueness of data, among other characteristics. In turn, these problems make the context of programmed decisions less relevant than its counterpart, the non-programmed decision. Where qualitative parameters compose the majority of executive decisions, [3], [5], [6], [15], [16], [20].

An important point that must be taken into account is also the culture of the enterprise, [4], [5], since it cannot be inherited, but is developed and is transferred by those who comprise the environment, that is, it is unique and innate for each organization. Decision can be seen as resolution, determination or deliberation, [10], to assume predictable risks with responsibility. According to Chiavenato, [3], decision is the action of analyzing alternatives and choosing one among the available ones, desirably, in real time, [20].

For William Starbuck, resident Professor of the Charles H. Lundquist College of Business, University of Oregon, “decision implies the end of deliberation and the beginning of action”, [14].

2.3 Information Systems

The quality of a decision is directly proportional to the trustworthiness of its data, [5], [21]. Stratification of information inside of an organization, [6], can be represented as the model shown in Fig. 1. In this model, quantitative information is gradually replaced by qualitative one and structured problems are also gradually replaced by unstructured ones, where impact of strategic decisions may be crucial to the fate of the business.

It is import to highlight that professional skill distribution within the organization is directly proportional to the intrinsic capacity of solving more complex problems and whose decisions present more risks for the survival of the enterprise as a whole,

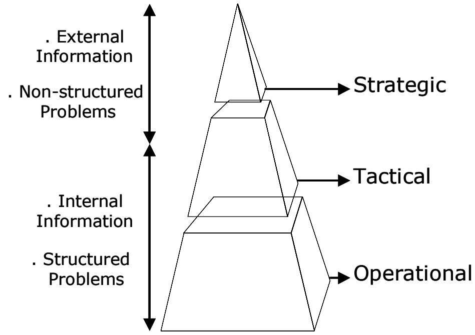


Fig. 1. Stratification of information within the organization (the tip of the tetrahedron represents the executive level)

[3],[6],[22]. This becomes easier to understand as we classify decisions in operational, tactical and strategic decisions in relation to their short-, medium- and long-term effects, respectively.

Products of information systems are always associated with decisions, [17], and for each level of information there are specific systems, as it follows:

- In the operational level, repetitive data are to be treated based on hard rules (i.e. transactional systems);
- In the tactical level, data synthesis deriving from transactional systems, such as report form, is usually the fundamental tool. Sometimes, some external information about clients, suppliers becomes necessary;
- In the strategic level, where decisions are predominantly non-repetitive and unpredictable, new approaches have to be considered regularly. In such level there are a few requirements to be observed, they are flexibility and adaptability in the presence of fast scenario changes, and desirably intuitive interface, graphic and textual support. These features guide the permanent search for modern and adaptive techniques to supporting executive decision making. Here is precisely where AISs can contribute significantly, [3], [16], [17], [23], [24], [25].

In general terms, information systems are developed to attend organizational objectives and are composed of input, transformation and output components. In this framework, a repository for storing-recovering data and information, models of representation of real problems, and an intuitive user interface are essential for its usability, [5], [17].

In this context, simple internal data and those which were transformed on information are considered as the knowledge of the enterprise about a specific segment, where reutilization of these stored information can be seen as specific memory lymphocytes that will perform quickly on future support for decision making, [17].

The main commercial techniques applied to implementations of decision support systems are the following: (*i*) statistical methods, (*ii*) decision tree, (*iii*) artificial neural networks and (*iv*) fuzzy logic. Choosing one of these methods depend, mainly, on the problem features. Specifically, fuzzy logic is the closest technique to our approach as they present some features centered in man. In general, the mentioned techniques are based on quantitative data used to construct executive indicators. Our approach uses qualitative data to directly generate appropriate decision cells. In this sense, there are not similar works with such a purpose.

3 The New Approach

3.1 Analogy Between Biological and Organizational Models

In this paper we introduce the concept of Immunedecision.[1] We define it as the set of appropriate responses for information that propitiate the 'decision-homeostasis' with

[1] Immunedecision is a neologism, created to better illustrate the hidden connection between immunology and decision theory.

in the enterprise. As decisions are composed by the following six elements, [3]: (*i*) the decision maker; (*ii*) decision objectives; (*iii*) preferences of the decision maker; (*iv*) decision strategy; (*v*) environmental aspects; (*vi*) decision consequences. Fig. 2 shows a macro-vision comparison of biological and organizational systems. We correlate them in order to draw some inspirations from biology. Notice that in both systems external inputs trigger internal responses. In the proposed approach the immune response, as pathogens arrive, is equivalent to the necessity of an appropriate decision.

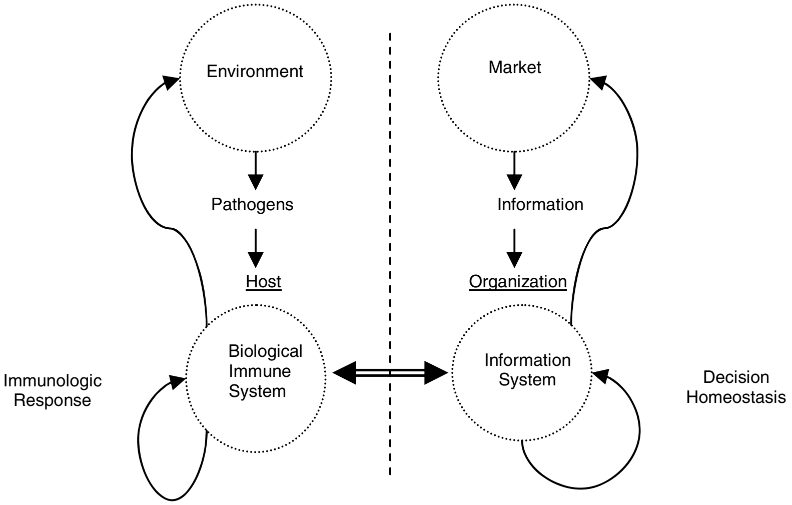


Fig. 2. Domains of the two identified analogous systems: (left) the biological and (right) the organizational

Further issues identified in the analogy between biological immune system and information systems and their fundamental characteristics are detailed as it follows:

- In biology as in organizations, systems work normally in harmonious, integrated and collaborative way;
- Immunologic responses (as executive decisions) are unique for each individual. Various factors such as environment and market are decisive in each case;
- Fungi, viruses, bacteria and parasites are pathogenic agents for the immune system, such as internal and external information are the external element that trigger decision making;
- Pathogenic agents instantiate immunologic actions, such as information instantiate decision actions, that is, homeostatic decisions;
- Lymphocytes, macrophages and leucocytes correspond to possible actions that can be carried out by directors, managers and supervisors;
- Phagocytosis correspond to the process of knowing, analyzing and decompose an information for future decision making;

- Immune system performs efficiently against infections avoiding possible harms, in the same way appropriate decisions may promote good consequences to the organization;
- Antigens can not trespass through the epidermis, the first defense barrier, in the same manner that external information that keep that way (*i.e.* that do not enter the organization) do not trigger any decision action;
- The work of the innate immunologic mechanism on phagocytosis can correspond to specific decisions for structured problems;
- Multiple antigens detections can occurs in the body simultaneously without the need of a central control mechanism. The same happens with simple decisions that can be made (*i.e.* detection, analysis and action) in any of the three levels of an organization structure;
- Detections based on affinity are carried out by the immune system based on a number of 'non-self' patterns if this number is greater than a specified threshold. This corresponds to decision making without complete knowledge about all needed aspects of the decision;
- Generation of diversity of lymphocytes receptors can be seen, in the organizational domain, as the use of scenarios creation techniques. This is of fundamental importance for supporting decisions;
- Immune system learns pathogens structures to propitiate future quick responses against these agents. The same happens as in new decisions opportunities, when the decision maker can make good use of new information and prior knowledge;
- As ability of recognizing 'self' and 'non-self' is vital in the immune system, selection of relevant information for solving a problem is also vital in the business world;
- Performance in response of the immune system is not attached to any particular process such as phagocytosis, affinity maturation, etc., but it is due to a well integrated sequence of processes. Decision responses in organizations are judge to obey the same rule;
- Transactional systems are intimately associated with the daily routines of an enterprise. This is believed to be similar in innate immune system.

3.2 Characteristics of the Proposed Model

The proposed model, introduced in this paper, aims at developing an appropriate information selection system for supporting executive decision making. The algorithm is based on the negative selection algorithm. This reunites AIS and decision theory; the previous section explains why we judge they are complementary.

Fig. 3 shows a model for the proposed architecture, where the repository database (DB) stores quantitative indicators of self performance for an enterprise, those, derived from transactional systems and external information.

A major problem that can be tackled by our approach is to minimize the undesirable fact that most other approach requires, that is need of complete information [5] [6]. Additionally, our approach tackles another difficult aspect of executive decision systems, that is, the ill-posed logic attached to those systems.

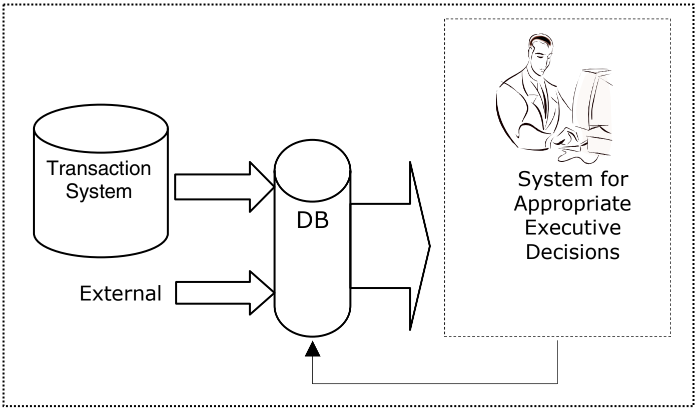


Fig. 3. Overview of the proposed Architecture

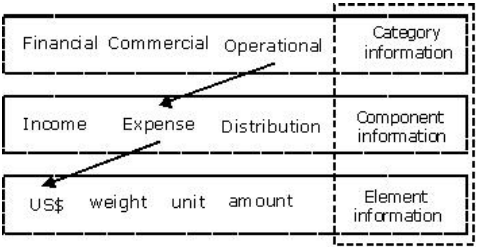


Fig. 4. Information encapsulation according to the hierarchy of concept

In Fig. 4, the encapsulation of information can be seen as a hierarchy of concepts, they are: category (Cg), component (Cp) and element (E_t) [26] [27]. According to [28], this principle is central to best representing information for decision making. We find that, together, these concepts are equivalent to the receptors of an antigen. Here we refer to them as 'decision receptors'. They are needed for the computation of affinity (refer to section 3.3).

We propose that the decomposition of a decision problem into its decision receptors is highly necessary for the mapping of solutions to the various decision cells (D_c). Each decision cell is then composed of *n* decision receptors. The higher the affinity with the problem the most receptors are matched. In Fig. 5 we shown the analogy of lymphocytes and decision cells (refer also to Fig. 4).

In the repository database (DB) are also stored collections of Cg, Cp, E_t and *decision memory* (D_m) that correspond to the decision cell approved by the decision maker.

Our proposed algorithm is made of two distinct stages (see Fig. 6). In the learning stage, strategic information is stored in DB (for instance: consuming market, suppliers, technology, etc.). With this information the decision maker will generate the first decision cells. Thpese cells make the initial repository of self cells for future generalizations that propitiate a secondary answer (similar to immunological system) in future company expositions to information that have originated these cells.

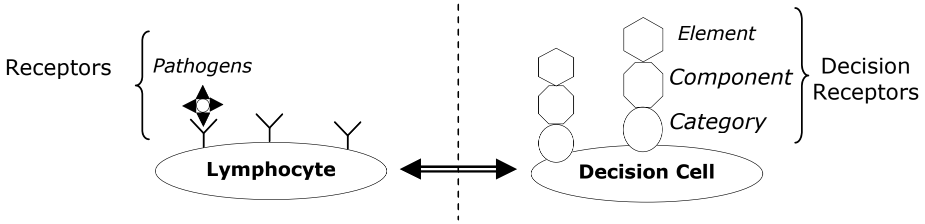


Fig. 5. Analogy between Lymphocyte and Decision cell (left and right, respectively)

In the operation stage, the decision maker requests to the system a decision cell of a specific type (for instance: to buy, to sell, to transform, etc.). After generation and presentation of the most appropriate cell, the decision maker may be requested (not always) to decide whether it is self or non-self. In the affirmative case, the generated cell is added to the pool of memory cells. Also at this stage, new cells of memory can be created.

The non-linear characteristic of the system emerges from diversity of stored decision receptors that propitiates a better generalization through the increasing number of memory cells.

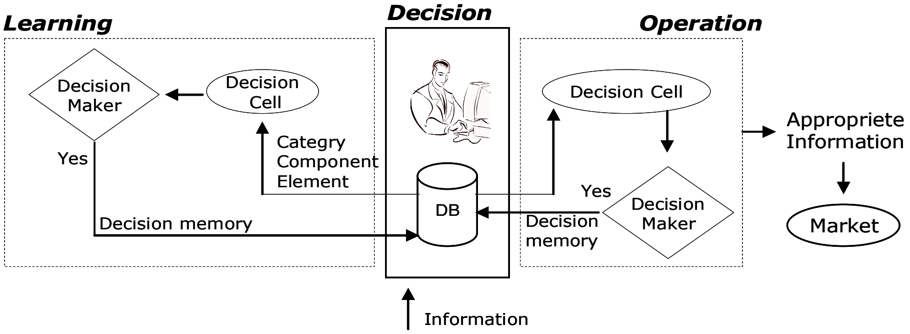


Fig. 6. Stages of the proposed system

3.3 Computations Within the Proposed System

The pseudo-code for execution of Appropriate Executive Decision (AED) is presented in Fig. 7 and its principle is based on the negative selection algorithm, modified for the real condition of work. Hamming distance is used to evaluate the affinity of the cells. Three parameters, predefined by the decision maker, are supplied at this stage for generation of cells, where:

- $\boldsymbol{a}$ – represents the type of cell that will be created;
- $\boldsymbol{\varepsilon}$ – represents the minimum affinity threshold that $\boldsymbol{n}$ cells generated randomly in the interval [1, $\boldsymbol{n}$], where **n** is equal to the number of memory cells, should have in relation to each original memory cell;

- $\boldsymbol{\varepsilon_c}$ – represents the minimum crossed affinity threshold that each cell should have in relation to other original cells;

The growth of $\boldsymbol{\varepsilon}$ represents a larger similarity between the memory cell and $\boldsymbol{n}$ generated cells, and the growth of $\boldsymbol{\varepsilon_c}$ represents the similarity level in relation to other memory cells. This strategy assures a larger reliability of the cell that will be presented to the decision maker, since $\boldsymbol{\varepsilon_c} < \boldsymbol{\varepsilon}$ In our proposal, we adopted the values of $\boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon_c}$ in the interval [1,10].

The function $\boldsymbol{r}$, given by (1), represents the reliability of the produced answer, defined in function of: $\boldsymbol{n_\varepsilon}$ - number of cells that attend $\boldsymbol{\varepsilon}$, and $\boldsymbol{n_{\varepsilon c}}$ – number of cells that attend $\boldsymbol{\varepsilon_c}$ ($\boldsymbol{n_{\varepsilon c}}$ is a subset of $\boldsymbol{n_\varepsilon}$); $\boldsymbol{r}$ varies in the interval [1-100] and represents the degree of the appropriateness of a candidate decision.

$$r = \exp \frac{-(n_\varepsilon - n_{\varepsilon c})^2}{2 * n_{\varepsilon c}{}^2} \qquad (1)$$

```
AED(a, ε, ε_C)//Appropriate Executive Decisions(AIS-based)
a: decision type
ε: affinity threshold
ε_C: cross affinity threshold
// Learning stage
While decision maker is not satisfied
   Input $n_m$ decision memory cells (by decision maker)
// Operation stage
While decision maker is not satisfied
    While $\varepsilon$ is not satisfied
        Generate $n_a$ decision cells in the interval $\{1, n\}$
    End While
    While $\varepsilon_c$ is not satisfied
        Present decision cell that satisfies $\varepsilon_c$
    End While
    Evaluation of $r$
    If decision cell is approved by decision maker
       Save decision cell as a decision memory
    End If
End While
```

Fig. 7. Pseudo-code for the AED-AIS Algorithm

4 Experiments and Results

To illustrate our approach we conceived one application where seven (actual) decision makers, responsible for purchases and sales contributed with their views on ‘how to decide about good bargains’. Based on that, we have created sound initial decision-cells, they can be seen in Table 1.

Table 1. Initial memory cells used as a basis for future purchases (they were provided by real people)

Decision Memory	Decision Receptors
D_{m1}	Price, stock, cash, quality, support
D_{m2}	Price, stock, cash, quality, period
D_{m3}	Price, stock, payment, quality, demand, period, supplying
D_{m4}	Price, stock, cash, climate, market, seasonality
D_{m5}	Price, stock, cash, profit, demand
D_{m6}	Price, warranty, service, quality, demand, period, support, attendance
D_{m7}	Cost, stock, cash, return, demand

The algorithm presented in the previous section was implemented and a simple graphic interface was prepared for testing our ideas. This application was utilized for the insertion of all decision receptors, of all 7 memory cells during the initialization of the system.

In the operation stage, the same decision makers (mentioned above) were invited to use the system, which has generated decision cells that could be approved or rejected by them. Table 2 contains results produced after intensive use of the system; each line is an example of a generated decision (cell) by the AED-AIS. Notice that their ‘receptors’ were not in any arrangement previously known. However, they are organized in such a way that, collectively, they are within boundaries of affinity that may be sound to executive users.

Three examples of generated decisions, presented in Table 2, may now be re-visited by the decision makers (*i.e.* they may be customized as appropriate decisions). The first one, despite the fact of a very low reliability, an actual decision maker considered the generated cell as appropriate. The second one, because of the increment of accepted cells, despite of the generated decision cell be of high reliability, the consulted decision maker is entitled to consider it as non-appropriate. Finally, in the third example, a cell of high reliability was generated and it was in deed accepted by the decision maker.

The column n_{memory}, represents the number of cells stored and competent for future decisions and the column "yes/no” represents the power of personal inference of the decision maker on the system.

Table 2. Results of generated decision cells after intensive use of the system

Decision Type	ε	ε_c	n_{memory}	n_{ε}	$n_{\varepsilon c}$	r (%)	Yes / No
Purchase	3	2	7	25	7	4	Yes
	Price, stock, cash, supplying, market, profit						
Purchase	7	3	8	21	17	97	No
	Price, stock, payment, quality, demand, period, supplying						
Purchase	3	1	8	32	27	98	Yes
	Price, stock, cash, market, service, seasonality						

The parameters presented in Table 3 were used in the experiments carried out for validation of reliability; results are presented in Fig. 8. Notice the slow decay of the reliability as cell numbers are increased.

Table 3. Parameters used experiment

Parameter	Value / range
a	Purchase
E	Set [2,7]
ε_c	Set [1,7] for $\varepsilon_c < \varepsilon$
Memory cells	[1,7] increased by interaction
Total of presented samples	111

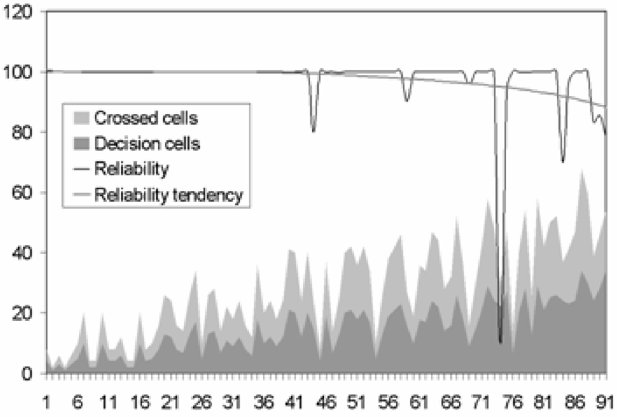


Fig. 8. Evolution of the reliability in function of cell number (x-coordinate is sample number; y-coordinate is reliability)

After analyzing the graph presented in Fig. 8, we highlight:

1. System reliability stays close of 100% at the major part of the experiment, except in intervals where the system does not generate crossed cells, that is, points with high values of ε e ε_c;

2. For values with reliability less than 100%, (e.g. sample 44, 59, 74) the number of crossed cells is smaller than of decision cells, ratifying the low reliability presented;
3. Between sample 1 and 90, number of memory cells were increased in steps of 1 unit. Notice that the growth of the number of memory cells increases the number of crossed cells and propitiate a larger reliability of the system. The growth in the number of memory cells acts like the secondary answer supplied by the artificial immunological systems.

5 Conclusion

This paper presented a new approach for selecting appropriate executive decision. The algorithm was inspired upon negative selection. A comprehensive analogy between human immune system and decision systems was presented as well.

The experiments show that the proposed intelligent algorithm is able to produce new instances of what we refer to as decision cells (in a clear reference to lymphocytes). The operation of the algorithm relies on a set of primordial decision that is kept as memory cells to guide the selection of new appropriate decisions. The reliability of generated decision cells, as an indicative of how appropriate information is, presented a marked increase growth as a function of number of memory cells. In addition to this, reliability values tune-up very quickly according to specific executive decisions domains. This is good because as we discussed earlier, decision maker preferences vary greatly.

We understand that some other aspects should be considered and further investigated. Namely, (*i*) how much learning of the system is dependent on the decision maker ability to provide good primes (*i.e.* on make good decisions); (*ii*) the possibility to gather good prototypical good decisions (as in a public repository for specific decision domains); (*iii*) how important is the number of memory cells to drive decisions in a more appropriate manner.

We conclude by highlighting that the fast adaptation and convergence of the proposed algorithm are strong indications that such a system can be modified to be used in selecting appropriate executive decisions in real organizational environments.

References

1. Castro, L.N., Von Zuben, F.: Artificial immune systems: Part I and II – a survey of applications. Technical Report DCA-RT 02/00, Department of Computer Engineering and Industrial Automation, School of Electrical and Computer Engineering, State University of Campinas, SP, Brazil (February 2000)
2. Castro, L.N.: Engenharia imunológica: desenvolvimento e aplicação de ferramentas computacionais inspiradas em sistema imunológicos artificiais. PhD thesis, Faculdade de Engenharia Elétrica e de Computação, UNICAMP, Campinas, SP, Brasil (2001)
3. Chiavenato, I.: Introdução à teoria geral da administração: uma visão abrangente da moderna administração das organizações, 7th edn. Ver. E atual, Rio de Janeiro. Elsevier, Amsterdam (2003)
4. Cruz, T.: Gerência do conhecimento. São Paulo, Cobra (2002)

5. Gomes, L.F.A.M., Gomes, C.F.S., Almeida, A.T.: Tomada de decisão gerencial, 2nd edn. São Paulo, Atlas (2006)
6. Stair, R.M.: Princípios de sistemas de informação: uma abordagem gerencial, seg edn. Rio de Janeiro, LTC (1998)
7. Dasgupta, D.: Advances in artificial immune systems. IEEE Computacional Intelligence Magazine (November 2006)
8. Correia, L.: Vida Artificial, XXV Congresso da Sociedade Brasileira de Computação, julho, UNISINOS – São Leopoldo/RS (2005)
9. Dasgupta, D., Gonzalez, F., Kozma, R.: Combining negative selection and classification techniques for anomaly detection, Computer science division, Mathematical Sciences department, The University of Memphis and Universidad National de Colombia, ©, IEEE (2002)
10. Ferreira, A.B.H.: Novo dicionário eletrônico aurélio versão 5.0 (2004)
11. Reis, M.A., Fernandes, D.A.M., Paula, F.S., Geus, P.L.: Modelagem de um sistema de segurança Imunológico (2007), http://www.doc.unicamp.br/~ra000504/papers/ImunoMod_SSI2001.pdf
12. Roitt, I.M.: Imunologia 5edn. Tradução Moysés A. Fuks. Atheneu Editora, São Paulo (1995)
13. Tierney Jr., L.M., McPhee, S.J., Papdakis, M.A.: e autores associados, Diagnóstico e tratamento: um livro médico Lange, Distúrbios alérgicos e imunológicos. Atheneu Editora, São Paulo (1998)
14. Buchanan, L., O'Connell, A.: Uma breve história da tomada de decisão, Harvard Business Review – Brasil (2007), http://www.hbrbr.com.br/textos.asp?codigo=10503
15. Almeida, A.T., Ramos, F.S. (org.): Gestão da informação na competitividade das organizações, Recife, Ed. Universitária (2002)
16. Audy, J.L.N., Brodbeck, A.F.: Sistemas de informação: planejamento e alinhamento estratégico nas organizações, Porto Alegre, Bookman (2003)
17. Shimizu, T., Neto, A.F.: Sistemas flexíveis de informação. Makron Books, São Paulo (1996)
18. Resende, S.O.: Sistemas inteligentes – fundamentos e aplicações, Barueri, SP, Manole (2005)
19. Kaufman, B.E.: Emotional arousal as a source of bounded rationality. Journal of Economics Behavior & Organization 38, 135–144 (1999)
20. Leigh, A.: Perfect decisions. Arrow Books (1993)
21. Inmon, W.H.: Como construir o data warehouse, tradução de Guz, A.M.N. Campus, Rio de Janeiro (1997)
22. Oliveira, D.P.R.: Sistemas de informações gerenciais: estratégicas, táticas operacionais. Atlas, São Paulo (1997)
23. Furlan, J.D.: Modelagem de negócio. Makron Books, São Paulo (1997)
24. MacGee, J., e Prusak, L.: Gerenciamento estratégico da informação: aumente a competitividade e a eficiência de sua empresa utilizando a informação como uma ferramenta estratégica. Campus, Rio de Janeiro (1994)
25. Turban, E.: Decision support systems and expert systems, 4th edn. Prentice-Hall International Editions, New Jersey, USA (1995)
26. Neto, F.B.L., Ludemir, T.B.: Decision support on sugarcane harvest (in Portuguese). In: IV Brazilian Symposium on Neural Networks – SBRN. Goiania, Brazil (1997)
27. Neto, F.B.L.: Managerial decision support, based on artificial neural networks (in Portuguese), Master Dissertation presented to Department of Informatics, Federal University of Pernambuco, Recife, Brazil (1998)
28. Oliveira, F.R.S., Pacheco, D F., Leonel, A., Neto, F.B.L.: Intelligent support decision in sugarcane harvest, Computers in Agriculture 2006, Lake Buena Vista, Florida, USA, (July 2006)

An Artificial Immune System-Inspired Multiobjective Evolutionary Algorithm with Application to the Detection of Distributed Computer Network Intrusions

Charles R. Haag, Gary B. Lamont, Paul D. Williams, and Gilbert L. Peterson

Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology (AFIT)
2950 Hobson Way, Bldg. 640
WPAFB, Dayton, OH, 45433-7765
charles.haag@langley.af.mil,
{gary.lamont,paul.williams,gilbert.peterson}@afit.edu

Abstract. Contemporary signature-based intrusion detection systems are reactive in nature and are storage-limited. Their operation depends upon identifying an instance of an intrusion or virus and encoding it into a signature that is stored in its anomaly database, providing a window of vulnerability to computer systems during this time. Further, the maximum size of an Internet Protocol-based message requires a huge database in order to maintain possible signature combinations. To tighten this response cycle within storage constraints, this paper presents an innovative artificial immune system (AIS) integrated with a multiobjective evolutionary algorithm (MOEA). This new distributed intrusion detection system (IDS) design is intended to measure the vector of tradeoff solutions among detectors with regard to two independent objectives: best classification fitness and multiobjective hypervolume size. AIS antibody detectors promiscuously monitor network traffic for exact and variant abnormal system events based on only the detector's own data structure and the application domain truth set. Applied to the MIT-DARPA 1999 insider intrusion detection data set, this new software engineered AIS-MOEA IDS called jREMISA correctly classifies normal and abnormal events at a relative high statistical level which is directly attributed to finding the proper detector affinity threshold.

1 Introduction

Signature-based intrusion detection systems (IDS) detect attacks by discovering exact matches between incoming data and an anomaly database of known attack string signatures. Their reactive nature and storage-limited database allows for example only a subset of the 256^{65535} harmful signature combinations to be catalogued for future detection after an attack. Compounding this situation is the trend of exact and variant nefarious signatures since 2001 reflecting exponential growth [15]. This situation helps to conceptually define an application domain known as the intrusion detection problem.

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 420–435, 2007.

Intrusion detection (ID) can be viewed as the problem of classifying network traffic as normal (*self*) or abnormal (*non-self*). Researchers in this area have developed a variety of intrusion detection systems (IDSs) based on statistical methods, neural networks, decision trees, and artificial immune systems (AISs). For example, various AIS approaches for IDS have been developed and evaluated, and are compared [9]. But, more effective and efficient IDS approaches are required. Our objective is to develop and evaluate a new innovative IDS approach integrating a multiobjective EA (MOEA) with contemporary AIS techniques. Performance of the resulting IDS is statistically addressed.

The intrusion detection method of choice is the artificial immune system (AIS) because the IDS architecture is similar to the human biological immune system (BIS): a parallel and distributed adaptive system [9] for detecting and destroying antigens. An AIS-based IDS attempts to classify network traffic as either *self* or *non-self* by evolving a population of antigen detectors (antibodies) based upon training data. A randomly generated set of immature detectors is created and exposed to normal network traffic (self) for a user-specified amount of time. If a possible detector (string) matches within some distance measure a *self packet* then the detector is removed; this is called *negative selection* [12]. If a detector matches a threshold number of *non-self packets* (antigen) during the learning phase as exposed to self and non-self traffic, it can be employed as an intrusion detector; this is called *co-stimulation*. Those detectors that receive co-stimulation are promoted to *memory detectors* in IDS *memory space* and are assigned a longer life time. Memory detectors that invoke stronger (*secondary*) responses are assigned longer life times when they match more memory detectors. Note that this describes a simplified AIS model. Many variations exist [9].

The IDS method of signature string evaluation is analogous to the consensus string problem for a specified error metric. A consensus string of a set by definition is based upon the consensus error that minimizes the sum of the distances between it and all other strings in the set [14]. This problem defines our IDS problem classification to be Nondeterministic Polynomial time (NP)-Hard [4]. This inability to find an optimal matching string in polynomial time possibility leads to the inability to solve ID problems in real-time IDS operations. Because signature-based IDSs are of this combinatoric nature are quite large, it becomes infeasible to find optimal solutions in polynomial time. Thus, one should consider a stochastic approximation approach as found in an evolutionary algorithm (EA) which possesses an ability to generate acceptable solutions in polynomial time. We integrate a multiobjective EA (MOEA) with contemporary artificial immune techniques resulting in an innovative and validated software engineered IDS [7]. The goal and objectives of this investigation are defined in Section 2, the IDS design in Section 3, with experimental results and analysis in Section 4.

2 Goals and Objectives

With an AIS architecture selected, a MOEA approach was chosen for IDS integration. The MOEA is appropriate because the consolidation of information into a single aggregated objective tends to lose data granularity and offer the decision-maker only one solution versus a set of trade-off solutions. The AIS-MOEA combined IDS reflects a new proof-of-concept algorithm that advances the existing work of two AIS-motivated algorithms: Edge, Lamont and Raines' retrovirus algorithm (REALGO) [5,10] for its ability to escape local optima, with regard to string matching, and Coello and Cortés' multiobjective immune system algorithm (MISA) [2] for its initial employment of an AIS in a multiobjective EA context.

Our *hypothesis* is that the conceptual integration of REALGO and MISA can evolve an improved AIS-inspired multiobjective evolutionary algorithm (MOEA) for intrusion detection [9,11]. The MOEA should provide a set of tradeoff solutions for the decision maker with regard to the measurement of two independent objectives that seek a global best: highest network traffic classification fitness and selection of MOEA detectors based upon best multiobjective hypervolume[1] performance. Thus, the specific goal is composed of two objectives:

1. *Attain the highest correct classification rate possible.* A heuristic-based positive value is assessed for any outcome, increasing the fitness score based upon the highest classification fitness. The higher a detector's effectiveness, the lower the value of this objective's sum score; that is, minimize the classification error rate.
2. *Maximize a detector's hypervolume size relative to the affinity threshold.* Research shows detector effectiveness is impacted by hypervolume of a particular size [1,18]. Detectors are desired that do not stray too far from the pre-determined *negative selection* affinity threshold; hence, lower deviation scores are desired.

The first objective seeks the highest detection and classification effectiveness rate of Intrusion detectors. The methodology can generate two types of errors: false-positives (referred to as Type-I) or false-negatives (referred to as Type-II) errors. False positives are declared conditions or findings that do not exist, such as indicating that a normal event as abnormal. False negatives are failures to recognize a condition that existed, such as declaring an abnormal event as normal. This results in unrecognized and uninhibited harm in a system. Higher scores resulting from false detections are heuristically determined based on the type and intensity of the detector's error; hence, we desire the lowest overall score possible for the detector population which translates into its highest effectiveness.

Regarding the second objective, the lower values are directly related to finding the smallest number of required detectors. Detector size, should not be too high as to react to normal traffic and not too low as to not react to abnormal traffic. Hence, in addition to classification fitness, we also desire a detector size deviation value as close to zero as possible.

To support the analysis and tradeoff of the two measurable objectives, the combined integration of REALGO and MISA into a Java environment based upon sound software engineering principles is required including the development of an associated distributed architecture.

By definition, multiobjective algorithms produce multiple solutions which may not be optimal for each individual objective [1]. By adjusting one solution for greater optimality, we risk decreasing the desired value of one or more other solutions. Thus, we desire a set or subset(s) of *nondominated solutions* through *Pareto Optimality* (*P**). A solution is considered *Pareto*-optimal (in a global-minimum context) if each

[1] Hypervolume: an experimental measure of the quality of Pareto front approximations [1,18].

of its objective's values is less-than-or-equal to all solution point objective values and it has at least one value, in any objective, smaller than any other solution point's value for that objective. This set of *Pareto*-optimal points represents the *Pareto Front* (*PF**) of fittest (*non-dominated*) detectors that is provided to the decision-maker. This set gives that person more options in selecting solution points for future ID domain employment based on the classification-to-hypervolume tradeoff depicted by the *true PF**. Note that a MOEA usually finds only an approximation to *PF**. Upon termination of the algorithm, two vectors should compose each objective's score for the set of surviving "acceptable" detectors in the entire population, where each detector is an approximate *P** solution.

3 High and Low Level Design

Our IDS algorithm is developed through the methodical integration of a foundational AIS framework and a generic MOEA structure with operator modification due to the architectures of REALGO and MISA [1,7]. The design approach should employ good software engineering practice. Timmis and De Castro define a standardized AIS framework where the engineered solution is *application domain*-specific (see Figure 1) [3]. This framework can be thought of as a layered approach. The basis for such an AIS begins with the pre-defined *application* (problem) *domain*, which governs the method of *representation*. Once chromosome data structure representation (e.g., bit string, real-valued vector, length, etc.) is decided, one or more affinity measures are used to quantify interactions of the system's elements; e.g., *Hamming distance* measurements applies to bit string representation while *Euclidian distance* is applied to real-valued vectors. The top layer, *immune algorithms*, encompasses those functions that govern the behavior (dynamics) of the system; e.g., method of mutation, selection, evaluation, etc. Addressing these layers leads to our engineered domain-specific solution.

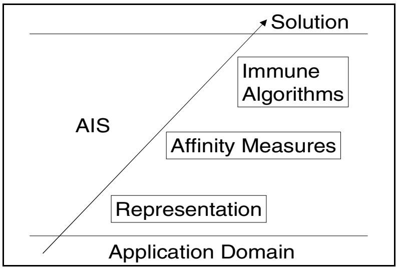


Fig. 1. Timmis' AIS framework [3]

3.1 Application Data Domain

The application domain is composed of week one and two of the MIT-DARPA (LL) 1999 insider intrusion detection data set [13] because it currently constitutes the largest publicly available benchmark of network traffic [8] and is normally used in the literature for evaluating IDS. The first week consists of normal (*self*)-only traffic, facilitating negative selection. The second week totals 7.2 million packets composed of 99.25% *self* and 0.75% labeled attacks (*non-self*). These sets are partitioned for training/learning and testing.

3.2 Antigen and Antibody Representation

The data structure that composes each *antigen* (Ag) data set record and *antibody* (Ab) detector chromosome is a fixed integer array with binary allele values that define the chromosome's location in the search space. This data structure was selected due to operator efficiency and moreover because it was (conveniently) the same type employed by both REALGO and MISA.

The Abs for network intrusion are generated and trained in the same manner as in anti-virus detectors [16]. However, network intrusion Ags are longer and segregated because they utilize the IP packet structure for its template. For this reason, we constrain our ID domain to encode Ags from network packets wrapped in the three most common IP protocols utilized by *non-self*: TCP, UDP and ICMP. Note that these protocols are embodied in the MIT-DARPA data. Decimal-value header information from each incoming Ag's packet header field is encoded into its binary equivalent and concatenated to the end of its string. Hence, each Ag's IP $\cup$ (TCP $\vee$ UDP $\vee$ ICMP) header results in a Ag TCP, UDP or ICMP binary DNA chromosome, respectively (see Figure 2). The intent is to employ all subfields within a specific protocol. All experiments performed involved all possible fields of the TCP, UDP, and ICMP headers, to fully evaluate the IDS algorithm's pattern-matching effectiveness. This means each TCP Ag was 240 bits, each UDP Ag was 170 bits and each ICMP Ag was 138 bits long.

From a microbiology perspective, Ab chromosomes by design are composed of three parts: its DNA (binary), RNA (binary) and seven state attributes (integer) (see Figure 3). Its DNA is generated by *negative selection* and the only portion of the Ab to be computed against the Ag. Its RNA is its DNA replica, facilitating the REALGO method of escaping local optima through *RNA reverse transcription* modeling [5,10]. If the mutated DNA results in a higher fitness than the previous time, its DNA is replicated to its personal memory space called RNA. If the fitness is worse, the DNA reverts to its last best fitness RNA to purpose mutation "in a different direction." Finally, there are seven parameters: λ ← name, α ← number of false detections, ρ ← (true positive + true negative) fitness score, ϕ ← (false positive + false negative) fitness score, η ← deviation from *negative selection* defined affinity threshold (determining volume), β ← broadcasted (yes/no), ψ ← number of Abs that *Pareto*-dominate this Ab.

Because our DNA is composed of a bit string data structure, the *affinity measure* of choice is the Hamming distance (Equation 1). Hamming distance is chosen over the r-contiguous bit rule, Euclidian, Manhattan, or other measures because the algorithm is pattern-matching the entire context of each Ag packet vs. particular contiguous IP fields [7,8].

$$H = \sum_{i=1}^{L} \delta_i \; where \; \delta_i = \begin{cases} 1 \; if \; Ab_i \neq Ag_i \\ 0 \; otherwise \end{cases} . \tag{1}$$

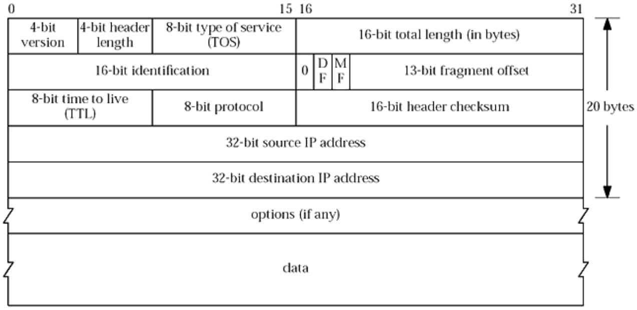


Fig. 2. Ag chromosomes are formed through IP (shown), TCP, UDP and ICMP packet header field decimal values

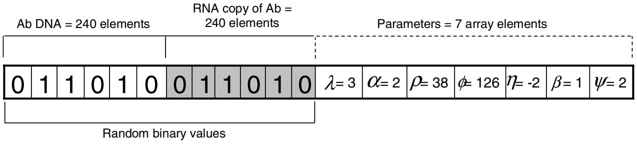


Fig. 3. Ab chromosome (DNA $\cup$ RNA $\cup$ parameters)

Distance measurements such as Hamming may not reflect the total semantical representation of IP data [6]. However, their use supports an efficient exploratory search of the fitness landscape and in our case the multiobjective landscape, attempting to generate “good” detectors of non-self.

3.3 Immune Algorithm

Integrating REALGO’s RNA transcription into the *evaluation* operator and the MISA framework, including its *evaluation*, *selection* and *mutation* operators, the pseudocode for our algorithm named jREMISA (*Java retrovirus-inspired MISA)* was developed as shown in Algorithm 1. Note that REALGO and MISA do not employ crossover because of the sufficiency of mutation to move Abs throughout the objective search space. Also, crossover can breakup good detector “building blocks.”

In understanding jREMISA, consider lines 3-7 of Algorithm 1, which consists of *negative selection* where Abs are randomly generated and evaluated against every Ag of a *self*-only day of traffic, given a predefined affinity threshold percentage. If an Ab reacts to *self*, it is discarded without replacement. In doing this, we are assured that the remaining population at algorithm termination does not react to a single packet of the day’s traffic. Post-*negative selection* consists of Algorithm 1, lines 8-19. These lines are reflected in the diagram of Figure 4, where each Ag entering the evaluation window represents a new generation. We partition our primary population (pop$_p$) by IP protocol for two reasons: efficiency is increased by evaluating the Ag only against a subset of pop$_p$ and pattern-matching becomes more relevant,; that is, TCP, UDP and ICMP do not all share the same chromosomal structure, as their IP fields differ; hence, it doesn’t make sense to compare a TCP Ab to a non-TCP Ag.

```
1   procedure jREMISA
2   begin
3   repeat
4       Randomly generate initial TCP, UDP, or ICMP Populations ($P_p$)
5       Initialize empty secondary Population ($P_s$)
6       negative_selection($P_p$,data_set$_{clean}$,threshold)      /* Evaluation 1 */
7   until (end of data_set$_{clean}$)
8   repeat
9       fitnessFunction (ag)                                 /* Evaluation_2 */
10      mutationCauchy($P_p$)
11      P_optimality()                                       /* Evaluation_3 */
12      clonalSelection(0.05)
13      mutationUniform($P_s$)
14      $P_p \leftarrow P_s$                  /* copy best of $P_s$ as next gen's $P_p$ */
15      if (networking)
16          broadcast($P_s$)               /* offer nondominateds to all AISs */
17          processReceived()                         /* Any captured Abs? */
18      endif
19  until (end of data_set$_{attack}$)
20  end
```

Algorithm 1. jREMISA AIS-inspired MOEA

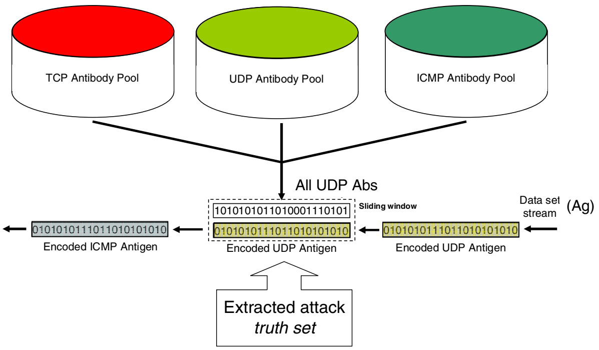


Fig. 4. Transient (data set) Ags evaluated against its IP protocol-matching Ab

When the Ag enters the window, the following process occurs (per jIEMISA, Algorithm 1):

- FITNESS FUNCTION (line 9): Hamming distance *H* is computed between the Ab and Ag DNA signature. Combined with the affinity threshold and *truth set* informing whether this Ag is *self* or *non-self*, one of four outcomes results:
  - True negative (Ag←Ab←*self*): penalize first objective (obj_1) fitness += *H*, copy DNA to RNA, reward second objective (obj_2) += 1%;

  - True positive (Ag←Ab←*non-self*): penalize obj_1 += (Ag_{length} – *H*), copy DNA to RNA, reward obj_2 += 1%;
  - False negative (Ag←self, Ab←non-self): falseDetections++, revert DNA to RNA, penalize obj_2 -= 1%;
  - False positive (Ag←non-self, Ab←self): falseDetections++, revert DNA to RNA, penalize obj_2 -= 1%.

Both true negative and positive outcomes are penalized because, if true negative, *H* should ideally equal zero. If not, obj_1's value is increased by the number of complimentary alleles (*H*). If true positive, then *H* should ideally equal the length of the Ag. If not, obj_2's value is increased by the number of non-complementary alleles;

- CAUCHY MUTATION (line 10): REALGO uses Cauchy mutation to move Abs around the landscape in a more exploratory mode than found with a Gaussian distribution. Mutation is performed on all penalized Ab alleles;
- *P**-TEST (line 11): all Abs undergo a Pareto-optimality test to determine the number of Abs each is dominated by. Quicksort then sorts them in ascending order using this number;
- MISA's CLONAL SELECTION PRINCIPLE (lines 12-14): elitist selection copies the top 5% of nondominated Abs from their pop_p to their respective secondary (or external) population (pop_s) meant to contain only nondominated Abs. Copied Abs are cloned to 600% the pop_s size. All copied and cloned Abs are then mutated in *n*-random allele positions, where *n* is the number of objectives (two) plus their *Pareto*-dominance value. Continuing, any pop_p Abs lost to reaching the max number of false detections is replaced by the fittest Abs from the pop_s in order to return the pop_p to its original size. Finally, the pop_s is culled for only nondominated Abs;
- AD-HOC NETWORKING (distributed, but optional): if enabled, newly discovered nondominated Abs copied to the pop_s are broadcast to the distributed subnet where listening jREMISAs capture and add to their pop_s only if it dominates their entire pop_s (see Figure 5). Since AIS detectors are rewarded for correct classification and detection even in a distributed IDS, an AIS can broadcast its detectors (fittest/elite, random, …) to the other AISs for possible inclusion into their local population. This process defines one generation and recurs for the number of data set packets. Various migration techniques can be employed [1].

3.4 Measures

The number of detectors generated depends upon the "optimal" affinity threshold found. Being multiobjective, each of our solutions (Abs) contains a set of two values:

1. an integer measure of how effectively they classify between self/non-self;
2. an integer measure of their affinity threshold (hypervolume) deviation from the starting affinity threshold defined at negative selection.

A global minimum is desired because: a) a higher fitness value means more penalties have been assessed for incorrect classifications; b) an Ab hypervolume should deviate as little as possible from the experimentally derived ideal affinity threshold of 39%

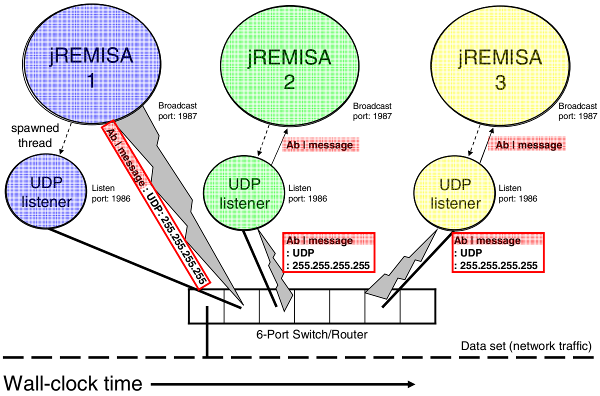


Fig. 5. jREMISA distributed communication architecture

[7]. The hypervolume Pareto compliant indicator is defined as the area of coverage of the known Pareto front with respect to the objective space for a two-objective MOP; that is, a hyperarea. This equates to the summation of all the rectangular areas, bounded by a user-defined reference point in objective space and points on the known Pareto front [1].

The software engineering of jREMISA in design and implementation used various paradigms; Model-View-Controller, UML Class Diagram, and Design Patterns [7]. jREMISA was developed in the *Eclipse*[2] open-source Integrated Development Environment (IDE) which provides a variety of excellent software design tools. This environment provides for ease of use with any application data set, selection of IDS algorithm parameter values, and visualization of IDS analysis.

4 Experimentation and Analysis

The Java development of the combined jREMISA Java package was evaluated over separate REALGO [6] and MISA [5] test functions which validated proper execution [4]. The design and implementation of a friendly user interface permits ease of use for application ID data representation, jREMISA parameter selection, experimental execution, visualization of results and analysis.

The detector training process uses the first two weeks of the 1999 corpus: the first week of self only traffic to negative-select our Abs and the complete second week of insider-only labeled attacks to develop the effectiveness of the trained Abs. The reality of jREMISA and based upon the intrusion detection objectives of Section 2, indicates that the evaluation experiments can easily be divided into two elements:

1. Compare jREMISA effectiveness measurements against the LL data set in 13 scenarios: 10 standalone, involving all days of week two, and 3 *distributed island model* executions in a two-, three- and four-jREMISA configuration;

[2] The Eclipse project, www.eclipse.org

2. Statistically compare jREMISA against another IDS algorithm applied to the same data set with experimental details known.

In the distributed configuration, four computers with Windows XP Professional 2002, Service Pack 2 were used:

- PC1: Dell Inspiron 710m laptop, 2 GHz Pentium-M, 2 GB of RAM;
- PC2: Dell XPS laptop, 3.4 GHz Pentium-4 HyperThreading, 1 GB RAM;
- PC3: Dell Precision laptop, 1.8 GHz Pentium-4, 512 MB RAM;
- PC4: Dell Optiplex GX270 tower, 2.6 GHz Pentium-4, 512 MB RAM.

Mapping the LL truth set of labeled-attacks to *Ethereal*[3]-analyzed five days of labeled attacks, 16 context-based attacks are extracted (Figure 6); attacks are fairly distributed in both time of day and day of week, varying in size of packets. Figure 7 indicates the trend of total event activity to non-self activity for each day.

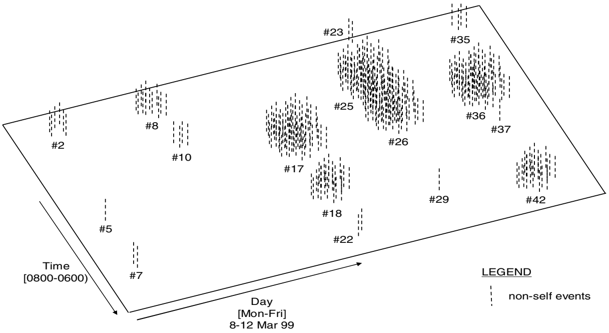


Fig. 6. LL 1999 week-two insider data set landscape with labeled attacks

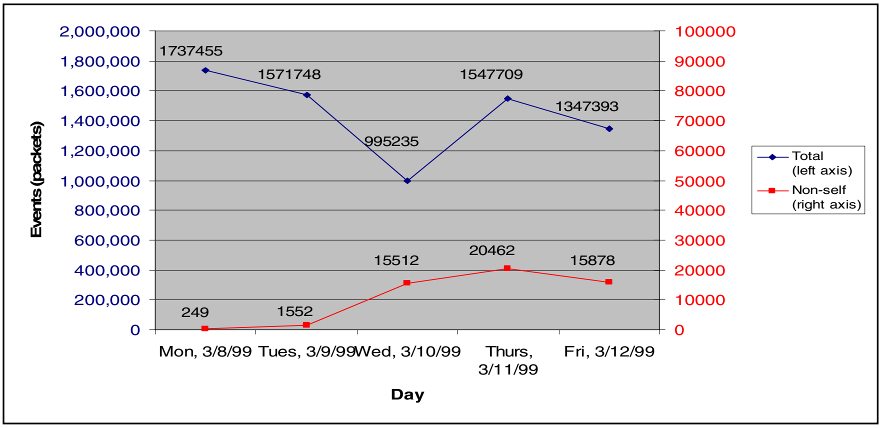


Fig. 7. MIT-DARPA "1999 week-two insider" landscape quantification

[3] Ethereal: open-source network protocol analyzer, www.ethereal.com

In comparing our results to Williams' *Warthog* or *CDIS* [16, 17], the surviving TCP pop_ps should be in the range of Warthog's experimentally-employed Abs: between 32 and 2048. Table 1 shows that this Ab range requires the affinity threshold between 37-42%. In Table 2, scenarios 1-6 show our Ab effectiveness based on each threshold. Since scenario 4 has the combined highest classification and lowest false detection rate, we use that 39% threshold to compare it to the remaining days of the week (scenarios 7-10). In Table 3, we employ this same threshold to the distributed tests, differing only by the number of partitions in the data set decomposition. Other researchers have also analyzed the MIT-DARPA (LL) 1999 insider intrusion detection data set with different IDSs, limited viewpoints, and lack of published experimental details. Thus, it is currently difficult to compare results.

Figure 8(a) represents scenario-four, TCP pop_s. The PF* is provided for all three populations for each MOEA execution. Its intent is to show the tradeoff between each Abs fitness score and affinity threshold deviation. For all five days of the week, all three pop_s *PF** had Abs in the +(4-5)% range 73% of the time. In addition, Abs are concentrated in this same range 87% of the time. Figure 8(b) shows an *attack graph* where each Ab plots its classification declaration for each attack. The x-axis indicates the outcome while the y-axis represents the packet number, mapping the attack. Here, there are no points on the false positive side for two attacks, indicating a 0% false positive rate for LL attack #26 and #29. While not shown, jREMISA also discovered LL attacks #7 and #22.

Table 1. Negative selection results for all popp starting at 4096 for Friday data set (1,467,775 packets)

Affinity (%)	Runtime (mins)	End TCP	survived	End UDP	survived	End ICMP	survived
37	186.65	2663	65.015%	3737	91.235%	3707	90.503%
38	124.20	1563	38.159%	3372	82.324%	3513	85.767%
39	89.17	935	22.827%	2890	70.557%	3290	80.322%
40	45.27	357	8.716%	2275	55.542%	2700	65.918%
41	26.43	126	3.076%	2000	48.828%	2344	57.227%
42	16.28	34	0.830%	1431	34.937%	1997	48.755%

Table 2. MOEA run summary: standalone jREMISA

								Self Events		*Non-self* Events	
Scen-ario	Day	Gener-ations	Affinity Threshold	TCP Pop	UDP Pop	ICMP Pop	Runtime	True Neg%	False Neg%	True Pos%	False Pos%
1	Thurs	1547710	42%	37	86	248	39.12 m	53.78	46.22	62.6	37.4
2	“	“	41%	106	116	284	52.48 m	67.44	32.56	68.33	31.67
3	“	“	40%	315	146	341	3.61 hrs	76.10	23.90	76.92	23.08
4	“	“	39%	966	361	810	18.21hrs	85.45	14.55	97.66	2.34
5	“	“	38%	1580	423	881	2.36 day	**86.48**	13.52	92.51	7.49
6	“	“	37%	2564	462	927	5.83 day	82.52	17.48	99.71	0.29
7	Mon	1737455	39%	969	349	846	20.02 hr	85.36	14.64	**99.90**	0.10
8	Tues	1571748	“	922	362	882	18.86 hr	84.61	15.39	97.35	2.65
9	Wed	995235	“	920	333	798	11.69 hr	83.37	16.63	98.26	1.74
10	Fri	1347393	“	964	376	829	13.43 hr	83.59	16.41	96.57	3.43

Table 3. MOEA run summary: distributed jREMISAs against Thursday data set

						Self Events		*Non-self* Events	
jREMISA ID	Packet range (1547709 total)	TCP Pop	UDP Pop	ICMP Pop	Runtime	True Neg%	False Neg%	True Pos%	False Pos%
Scenario 11: 2 jREMISAs, 39% affinity threshold, Thursday attack data set									
PC1	1 – 773854	966	361	810	9.44hr	**86.21**	13.79	98.10	1.90
PC2	773855 – 1547709	936	344	854	9.63hr				
Scenario 12: 3 jREMISAs, 39% affinity threshold, Thursday attack data set									
PC1	1 – 515903	966	361	810	5.09hr	84.31	15.69	97.94	2.06
PC2	515904 – 1031807	936	344	854	6.35hr				
PC3	1031808 – 1547709	951	357	826	6.86hr				
Scenario 13: 4 jREMISAs, 39% affinity threshold, Thursday attack data set									
PC1	1 – 386927	966	361	810	4.33hr	84.94	15.06	**98.55**	1.45
PC2	386928 – 773854	936	344	854	4.63hr				
PC3	773855 – 1160781	951	357	826	4.86hr				
PC4	1160782 – 1547709	954	360	822	5.09hr				

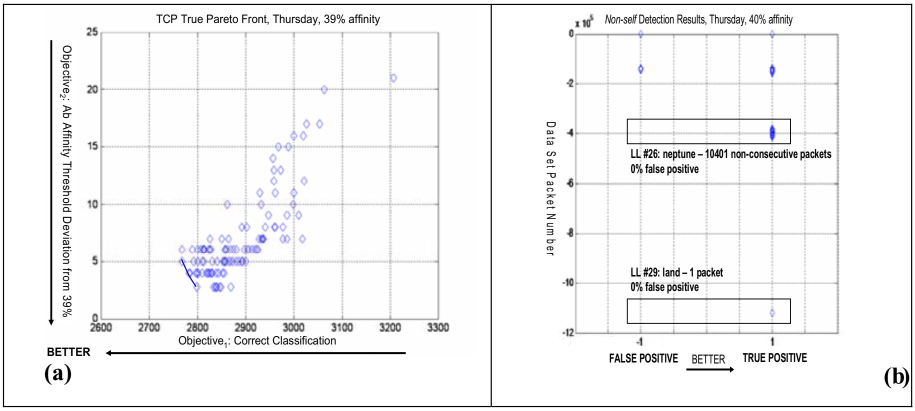


Fig. 8. MOEA post-execution: (a) TCP pops PF*; (b) attack graph: two attacks found with 0% false positive rate

Figure 9 graphically depicts Table 2 with classification and false detection ratios having 5% variance due to the multiple trials run of the same scenario to maximize statistical accuracy. Figure 10 and Figure 11 summarize the results of the distributed phase of our experiments. In Figure 10, a graphical depiction of Table 3, one discovers that the sharing of nondominated Abs among jREMISAs did not conclusively show a synergistic increase in effectiveness when compared to the standalone classification ratio, although the two-PC configuration fared better. Figure 11 shows the increase in efficiency as more jREMISAs participated. Upon the

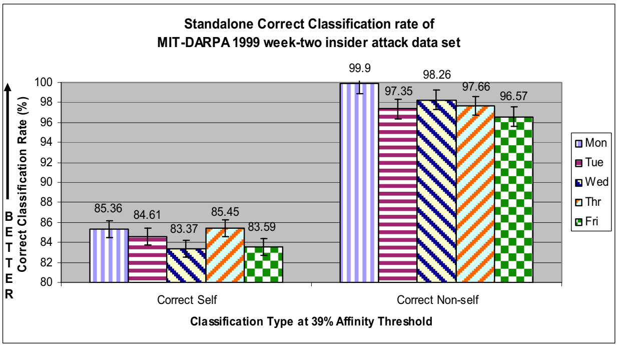


Fig. 9. jREMISA standalone effectiveness against each day of the week-two insider attack data set

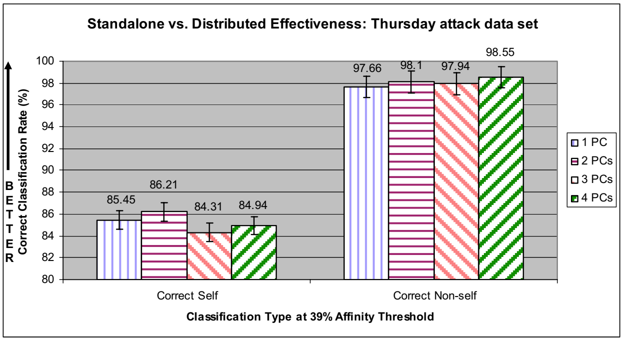


Fig. 10. Standalone vs. distributed effectiveness against the week-two Thursday insider attack data set

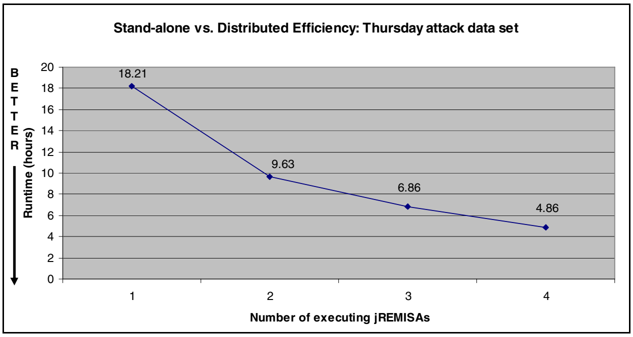


Fig. 11. Data decomposition-based distributed execution: efficiency vs. number of executing jREMISAs

completion of either phase—*negative selection* or MOEA—jREMISA saves the resulting population, *Pareto Front* and attack graph values to a formatted XML file.

Figure 12 shows how our algorithm does better than Warthog with regard to the false detection rate as the number of Abs are increased for the week-two Thursday data set. With Warthog, as the number of Abs increases, so does the false positive rate. However, our experiments show that as the Abs are increased, the false positive rate decreased. Therefore, jREMISA is relatively more effective than Warthog with respect to the same benchmark data set processed in a similar fashion.

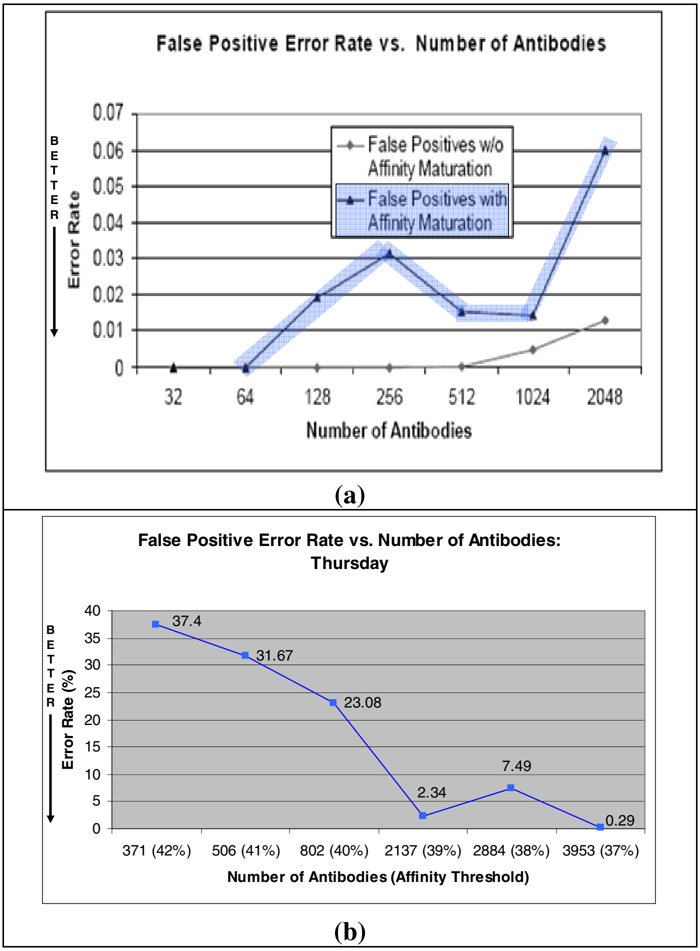


Fig. 12. False positive error rate trend: (a) Warthog vs. (b) jREMISA

5 Conclusion

REALGO and MISA were successfully migrated to Java and integrated via software engineering principles into jREMISA, a distributed computational IDS system. The two IDS performance optimality objectives (classification, hybervolume) and thus the design goal were validated based on jREMISA experimentation. This integrated algorithm achieved an average 83.37-85.45% *self* classification and 96.57-99.90% *non-self* classification rate for a 39% affinity threshold. We observed a patterned Ab hypervolume between 0-5% above this threshold (making their hypervolume 39-44%) and Ab broadcasting and receipt. In addition, jREMISA performed better in at least one way over another using the same data set. It detected four attacks, ranging from one to 10401 non-consecutive packets, with a 0% false positive rate. From this successful effort, the jREMISA IDS development is presented as an initial attempt and possibly the first at attempting to validate an AIS-inspired MOEA applied to the ID problem domain. The use of MOEAs shows promise regarding improvement to contemporary AIS IDS via performance tradeoffs.

By design, jREMISA is advocated as an open-source software engineered IDS permitting collaboration, improvement, modification, and multiple data set testing, with performance analysis tools. With appropriate experimental details forthcoming from other IDS investigations (using for example an AIS, Neural Network, SVN, Bayes decision tree, k nearest neighbor approach, …), comparison to other results with the MIT-DARPA data sets is suggested. Other jREMISA possibilities include completing the KDD Cup 99 data set facilitation and analysis, integrating Ab self/non-self geometric hypershapes, exploring advanced negative selection methodologies, using and comparing other affinity measures with jREMISA, adding more real-world objective functions and constraints, and analyzing performance evaluation using other MOEA metrics.

References

[1] Coello, C., Van Veldhuizen, D., Lamont, G.: Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer 2002, 2nd edn. Springer, Heidelberg (2007)

[2] Coello, C., Cortés, N.: Solving Multiobjective Optimization Problems Using an Artificial Immune System. Genetic Programming and Evolvable Machines 6, 163–190 (2005)

[3] De Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, London (2002)

[4] Dréo, J., Pétrowski, A., Siarry, P., Taillard, E.: Metaheuristics for Hard Optimization: Methods and Case Studies. Springer, Germany (2006)

[5] Edge, K., Lamont, G., Raines, R.: A Retrovirus Inspired Algorithm for Virus Detection & Optimization. In: Genetic and Evolutionary Computation Conference (GECCO '06) (2006)

[6] Gonzalez, F., Dasgupta, D., Gomez, J.: The Effect of Binary Matching Rules in Negative Selection. In: Genetic and Evolutionary Computation (CEC'03), Springer, Heidelberg (2003)

[7] Haag, C.R.: An Artificial Immune System-inspired Multiobjective Evolutionary Algorithm with Application to the Detection of Distributed Computer Network Intrusions. M.S. Thesis, Graduate School of Engineering and Management, Air Force Institute of Technology, WPAFB, Dayton, OH, (March 2007)

[8] Harmer, P., Williams, P., Gunsch, G., Lamont, G.: An Artificial Immune System Architecture for Computer Security Applications. IEEE Transactions on Evolutionary Computation 6(3) (June 2002)

[9] Kim, J., Bentley, P., Aickelin, U., Greensmith, J., Tedesco, G., Twycross, J.: Immune System Approaches to Intrusion Detection - A Review, Natural Computing. Springer, Heidelberg (2007)

[10] McGee, P.: Building Better Antibody Therapeutics, Drug Discovery & Development, www.dddmag.com/ShowPR.aspx?PUBCODE=090&ACCT=1600000100&ISSUE=0701 &RELTYPE=DEV&PRODCODE=00000000&PRODLETT=AG&CommonCount=0.

[11] Michalewicz, Z., Fogel, D.: How to Solve It: Modern Heuristics, 2nd edn. Springer, Heidelberg (2004)

[12] Middlemiss, M.: Positive and Negative Selection in a Multilayer Artificial Immune System. Information Science Discussion Paper Series, No. 2006/03, University of Otago (January 2006)

[13] MIT Lincoln Laboratory–DARPA Intrusion Detection Evaluation: www.ll.mit.edu/IST/ideval/

[14] Sim, J.S., Park, K.: The Consensus String Problem for a Metric is NP-Complete. J. of Discrete Algorithms 1(1), 111–117 (2003)

[15] Symantec Internet Security Threat Report; Trends for January 1, 2004 – June 30, 2004, vol. VI, (September 2004), eval.veritas.com/mktginfo/enterprise/white_papers/ent-whitepaper_symantec_internet_security_threat_report_vi.pdf

[16] Williams, P.: WARTHOG: Towards a Computer Immune System for Detecting "Low and Slow" Information System Attacks, M.S. Thesis, Graduate School of Engineering and Management. Air Force Institute of Technology, WPAFB, Dayton, OH, (March 2001)

[17] Williams, P., Anchor, K., Bebo, J., Gunsch, G., Lamont, G.: CDIS: Towards a Computer Immune System for Detecting Network Intrusions. In: Lee, W., Mé, L., Wespi, A. (eds.) RAID 2001. LNCS, vol. 2212, Springer, Heidelberg (2001)

[18] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Fonseca, V.G.d.: Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE Transactions on Evolutionary Computation 7, 117–132 (2003)

Author Index