AISB 2004 Convention:

Motion, Emotion and Cognition

AISB The Society for the Study of Artificial Intelligence and the Simulation of Behaviour

Proceedings of the AISB 2004

Symposium on The Immune System and Cognition

29 March – 1 April, 2004

ICSRiM, University of Leeds, Leeds LS2 9JT, UK

www.leeds.ac.uk/aisb www.icsrim.org.uk

AISB 2004 Convention

29 March – 1 April, 2004
ICSRiM, University of Leeds, Leeds LS2 9JT, UK
www.leeds.ac.uk/aisb www.icsrim.org.uk

Proceedings of the AISB 2004 Symposium on The Immune System and Cognition

Published by

The Society for the Study of Artificial Intelligence and the Simulation of Behaviour

http://www.aisb.org.uk

ISBN 1 902956 38 2

Contents

The AISB 2004 Convention

On behalf of the local organising committee and all the AISB 2004 programme committees, I am delighted to welcome you to the AISB 2004 Convention of the Society for the Study of Artificial Intelligence and the Simulation of Behaviour (SSAISB), at the University of Leeds, Leeds, UK.

The SSAISB is the oldest AI society in Europe and it has a long track record of supporting the UK AI research community. This year, the underlying convention theme for AISB 2004 is "*Motion, Emotion and Cognition*", reflecting the current interest in such topics as: motion tracking, gesture interface, behaviours modelling, cognition, expression and emotion simulation and many others exciting AI related research topics. The Convention consists of a set of symposia and workshop running concurrently to present a wide range of novel ideas and cutting edge developments, together with the contribution of invited speakers:

- Prof Anthony Cohn
  Cognitive Vision: integrating symbolic qualitative representations with computer vision;
- Prof Antonio Camurri
  Expressive Gesture and Multimodal Interactive Systems;
- Dr David Randell
  Reasoning about Perception, Space and Motion: a Cognitive Robotics Perspective; and
- Dr Ian Cross
  The Social Mind and the Emergence of Musicality,

not to mention the many speakers invited to the individual symposia and workshop, who will made the Convention an exciting and fruitful event.

The AISB 2004 Convention consists of symposia on:

- Adaptive Agents and Multi-Agent Systems;
- Emotion, Cognition, and Affective Computing;
- Gesture Interfaces for Multimedia Systems;
- Immune System and Cognition;
- Language, Speech and Gesture for Expressive Characters; and the
- Workshop on Automated Reasoning.

The coverage is intended to be wide and inclusive all areas of Artificial Intelligence and Cognitive Science, including interdisciplinary domains such as VR simulation, expressive gesture, cognition, robotics, agents, autonomous, perception and sensory systems.

The organising committee is grateful to many people without whom this Convention would not be possible. Thanks to old and new friends, collaborators, institutions and organisations, who have supported the events. Thanks the Interdisciplinary Centre of Scientific Research in Music (ICSRiM), School of Computing and School of Music, University of Leeds, for their support in the event. Thanks to the symposium chairs and committees, and all members of the AISB Committee, particularly Geraint Wiggins and Simon Colton, for their hard work, support and cooperation. Thanks to all the authors of the contributed papers, including those which were regretfully not eventually accepted. Last but not least, thanks to all participants of AISB 2004. We look forward to seeing you soon.

Kia Ng
AISB 2004 Convention Chair
ICSRiM, University of Leeds,
School of Computing & School of Music,
Leeds LS2 9JT, UK
kia@kcng.org www.kcng.org

Proceedings of the AISB 2004 Symposium on the Immune System and Cognition (ImmCog)

Symposium Preface

Most people only think about their immune system when they get sick, if then, and almost certainly they would not consider it to have the cognitive properties usually associated with the brain. Yet, the immune system is one of the most complex, intricate and robust systems known, consisting of hundreds of billions of elements, interacting in a elegantly choreographed dance. It may be seen as receiving sensory input (the detection of antigen), the ability to process that input (an immune response) and the ability to learn (improved response over time). Is this a fair assessment? If so, are there other aspects of the immune system that can be seen as cognitive?

This symposium is designed to explore this comparison to cognition, and it is hoped that the papers presented will spark discussion, debate and new possibilities. As a result, some of the papers have already been presented elsewhere, and some are only presented as abstracts; the aim is that the main ideas should be drawn together here in Leeds, and then, perhaps, the results published in an edited volume. The work presented here is as follows.

Jon Timmis provides a broad discussion of the view that the immune system can be seen as a cognitive system. He outlines the perceptive and active elements of the immune system, and sees a clear link between the immune system and the nervous system and brain. Jerne's theory of immune memory is also discussed. He concludes that the cognitive elements of this immune system can be summarised in properties: 'The search for a context', which dictates when it will act; 'Signal extraction from noise', which determines how to focus recognition; and 'The response problem', which decides what action to take.

Mark Neal also investigates immune cognition in general. He notes there are various views about what cognition is, and suggests that we should stretch existing definitions of cognition to cover the immune system. He then argues for a more holistic view of the biological systems that inspire Artificial Immune Systems (AIS), reminding us that most natural systems are concerned—to some degree at least—with homeostasis. As an example, he discusses the use of AIS in autonomous mobile robotics. Alternatively, Andrew Webb, Emma Hart, Peter Ross and Alistair Lawson, demonstrate the cognitive ability of the immune system by using and AIS to control a robot in such a way that it develops abilities in a manner analogous to human children.

Andrew Watkins and others focus on the memory aspects of cognition. Watkins investigates the AIRS system that models immune memory by the interactions between antigen and B-cells. William Wilson and Simon Garrett take a more general view of immune memory and report some initial work that sets out to produce a general model of immune memory that is useful for prediction in immunology, and which displays new memory dynamics that can be abstracted for use in AIS. Alessio Gaspar and Beat Hirsbrunner investigate the use of the immune system's secondary response to an infection—i.e. its memory of a past event—to improve reaction to previously encountered situations in a cyclic, continuous learning environment.

Jamie Twycross' short paper outlines the aims and scope of work, just begun, into the use of danger theory (DT). He introduces the concept of DT, a proposed detection/sensory mechanism that triggers immune responses. The main aim is to detect the misuse of computer networks and other systems in real-world environments. Two poster presentations also discuss the use of danger theory. Anjum Iqbal and Mohd Aizaini Maarof discuss the possibility of using antigen presenting cells (APC) to help mediate danger detection in the danger theory. Slaviša Sarafijanović and Jean-Yves Le Boudec propose an application of AIS for the detection of misbehaving nodes in

mobile, ad hoc networks. Their work draws on the principle of the danger signal and uses this to help to maintain quality of service in the network. I hope you enjoy the symposium and find that the discussions stimulate new ideas that you can pursue in your research.

Simon Garrett, Aberystwyth, March 2004

The Immune System, Networks and Cognition: Enough Thoughts for a Debate?

Jon Timmis
Computing Laboratory
University of Kent
Canterbury. Kent. UK.
J.Timmis@kent.ac.uk

This paper presents a review of contributions to the view that the immune system is somehow cognitive. There have been a number of views expressed in the immunology literature about this subject. Indeed when one begins to examine the literature, you can see that this idea has been round for a time, and in some places is growing in favor with immunologists and computer scientists (de Castro and Timmis, 2002). The main obstacle may be one of our perceptions of what cognition really is. Typically, the term is associated with the brain and the area of psychology. In this domain, cognition refers to the superior functions of the brain, such as object recognition, identification of the organism, and intentionality (Mitchison, 1994). Is it possible to argue such a viewpoint for the immune system? That somehow there are functions going on that 'do the same thing' albeit with different stimuli. This is an interesting question and debate. The objective of this paper is not to answer the question, but to highlight arguments that are in the literature and raise this question for the current research community to think about and maybe, this will lead to a new understanding of the immune system and then affect the area of AIS in new and interesting ways.

When the immune system is viewed as a cognitive entity, it represents a complement to the nervous system. For the immune system to act, it first has to perceive, recognise, and decide what mechanisms to put into action in order to operate. It can thus be inferred that it performs a sort of cognitive function, where the immune and nervous systems are viewed analogously. Both systems have perceptive properties: the capability of distinguishing between the internal and external universes. Information processing is central for their functioning and the respective perceptive properties are linked to effector mechanisms. Besides the functional analogies, the increasing evidence of their interdependence, through messenger molecules, neurotransmitters, and hormones was discussed previously. With the adoption of the immune network paradigm proposed by Jerne (1974), the similarities between these systems are even more striking.

Work by the well respected immunologists such as I. Cohen, F. Varela, A. Coutinho, and N. Jerne deal with *immune cognition* as based upon the self/nonself discrimination paradigm and/or the immune network theory. Whilst the earlier Jerne immune network theory has been mostly cast aside by immunologists, the idea that the immune system is now large network of interacting agents is still very much a popular point of view –therefore, many of these arguments still stand. This earlier work attempted to emphasise that the immune system in some way *knows* what it is looking for when it encounters an antigen, i.e., its internal organisation endows it with certain intentionality. The identification of foreign elements to the organism implicitly requires that some immune component is performing this identification, or recognition. Recognition is a *perceptive* event (registration of sensory stimuli) and, thus, has to be sustained in some sort of cognitive apparatus (Tauber, 1997). This standpoint reflects the richness hidden in terms like recognition, learning, and memory, properties pertinent to the immune system.

N. Jerne with his network theory is considered to be the true author of the cognitive model of the immune system (Tauber, 1997). The cognitive view of the immune network theory, has two underlying assumptions:

- The immune system is composed of a universe of internal images that are only recognised because they are expressed in a language known to the system; and
- The immune system is self-defined, i.e., it is designed to know itself.

This is of course is contrary to the danger theory proposed in (Matzinger, 1994), but that debate is left for another paper. Within the immune network, the self-elements promote a certain pattern of response, while the nonself induces another type of response. This is based not in the intrinsic nature of the nonself, but in the fact

that the immune system perceives the foreign antigen in the *context* of invasion or degeneracy. The key elements are the antibodies that act as antigens through their Idiotypic domains, thus existing an internal image of the universe of antigens. The mutual recognition among the immune components (B-cells and antibodies) form a large interconnected network, the immune or Idiotypic network, of elements that communicate with each other.

Following this same contextualist approach to the immune cognition, I. Cohen (1992) defined a cognitive system as one capable of extracting information and fashioning experience out of raw input data by deploying information already contained in the system. Thus, a cognitive system acts through a sense of direction with intentionality, it is not a passive information processor, or a memory of information; it is designed to manipulate particular information from the domain in which it operates.

Antigens are recognised as nonself because they are presented in a context that indicates their pathology – they are causing damage to the host organism. Autoimmunity is viewed as a normal characteristic of the immune system that constantly tries to identify and monitor the elements of the host. If these self-antigens are altered in a contextual form, their meanings change, and an immune response might be triggered. Hence, self is no longer an entity; rather it emerges dynamically in a self-identification process that changes continuously along the lifetime of an individual.

Similarly, a theory based on the definition of self and the immune network hypothesis was developed by A. Coutinho and his collaborators (Varela *et al.*, 1988; Coutinho, 1989; Bersini & Varela, 1990). They suggested that the global properties of the immune system such as self/nonself discrimination and self-tolerance couldn't be understood through the analysis of individual components. They proposed that essential properties of immune networks such as structure, dynamics, and metadynamics, together with clonal selection, constitute powerful approaches for the study of specific cognitive aspects of the immune system, such as recognition, learning, and memory. Immune memory was assumed to be a clonal characteristic, at least in the context of secondary responses, and antigenic recognition (directly related to memory) is probably the most appealing cognitive immune property. They classified the immune system as belonging to a class of biological systems whose adaptability relies on a continuous generation of novel elements (cells and molecules) to handle an unpredictable and varying set of situations (antigens).

Based upon the immune network models proposed by Varela *et al*. (1988) and the study of selective theories, Manderick (1994) discussed how selectionism could be applied to an understanding of autonomous cognitive systems. He argued that the adaptability principles incorporated by evolutionary systems, such as ecosystems and the immune and nervous systems (under the theory of neuronal group selection, TNGS, proposed by Edelman in 1987), are crucial for the understanding of cognitive behaviors. As pointed out by Varela and his collaborators (Varela *et al.*, 1988), he stressed as the main cognitive properties of the immune system its pattern recognition, learning, and memory capabilities.

As a conclusion of the several approaches to the immune cognition discussed above, it is important to remark that cognition in the immune system also implies consciousness: the properties of intentionality and personality. Both are unique and depend on the history and individual experiences of each organism (Tauber, 1994). Cognitive principles embodying the ideas of intentionality and symbolic manipulation (or computation) can be generically applied to the immune and nervous systems; both deal with:

- *The search for a context*: when to act;
- *Signal extraction from noise*: how to focus recognition; and
- *The response problem*: what decision to take.

This paper has tried to highlight a number of arguments within the immunological literature that relate to the idea of the immune system being a cognitive system. When viewed as a complement to the nervous system, this view maybe becomes a little more palatable. While not answering any questions, it is hoped this paper can act to fuel the debate on this issue.

Bersini, H. & Varela, F. J. (1994), "The Immune Learning Mechanisms: Reinforcement, Recruitment and Their Applications", In *Computing with Biological Metaphors*, R. Paton (ed.), Chapman & Hall, pp. 166-192.

Cohen, I. R. (1992a), “The Cognitive Principle Challenges Clonal Selection”, *Imm. Today*, 13(11), pp. 441-444.

Coutinho, A. (1989), “Beyond Clonal Selection and Network”, *Imm. Rev.*, 110, pp. 63-87.

De Castro, L.N and Timmis, J (2002). Artificial Immune Systems: A New Computational Intelligence Approach. Springer-Verlag.

Jerne, N. K. (1974), “Towards a Network Theory of the Immune System”, *Ann. Immunol.* (Inst. Pasteur) 125C, pp. 373-389.

Manderick, B. (1994), “The Importance of Selectionist Systems for Cognition”, In: *Computing with Biological Metaphors*, R. Paton (ed.), Chapman & Hall.

Matzinger, P (1994) A Renewed Sense of Self. Nature.

Mitchison, N. A. (1994), “Cognitive Immunology”, *The Immunologist*, 2/4, pp. 140-141.

Tauber, A. I. (1997), “Historical and Philosophical Perspectives on Immune Cognition”, *Journal of the History of Biology*, 30, pp. 419-440.

Varela, F. J., Coutinho, A. Dupire, E. & Vaz, N. N. (1988), “Cognitive Networks: Immune, Neural and Otherwise”, *Theoretical Immunology*, Parte Dois, A. S. Perelson (ed.), pp. 359-375.

What's most interesting: cognition, homeostasis or autonomy?

Mark Neal,
Department of Computer Science,
University of Wales, Aberystwyth,
Ceredigion,
U.K.
email: mjn@aber.ac.uk

Where's this heading?

The nature of cognition and the systems which are said to posess it have been a basic theme of artificial intelligence and philosophy for a long time, and we are not really any closer to definitive descriptions of mechanisms which are said to be cognitive than we were thirty years ago (arguably much longer). The fact that we are now willing to label systems like the immune system as cognitive can be seen as indicative of one of several things:

> I am wrong and we have identified the essential components of cognition and the immune system contains these components
>
> We have some intuitive understanding of what cognition is and feel comfortable about calling the immune system cognitive
>
> We are gradually stretching the definition of cognition to include systems that we would not previously have considered to be cognitive
>
> We are clutching at straws in our search for biological metaphors to exploit

All of these standpoints are propounded by various workers.

The definition of cognition is beyond the scope and interest of the author, but I would note that the author generally believes that the third item in the list is the case. On more charitable occasions he may lean more towards item two.

The aim of this paper is to return to the question of what we are really interested in when using biologically motivated computational systems, and in particular artificial immune systems.

What does the immune system do for us?

Simple reconsideration of the functions of the human immune system (in common with various other physiological system) brings to mind a single *fundamental* function:

> Survival

and a number of mechanisms which help to acheive this:

> Identification of threats
>
> Elimination of threats
>
> Ability to extend the threat repertoire

Functions identified at this level are independent of specific mechanisms and theories about how the immune system works (perhaps the word "threat" might be seen as an endorsement of the Danger Theory, but it is not intended to and in the absence of a completely theory-neutral word will continue to be used). Focussing on the immune system's ability to identify and eliminate threats leads to the idea of using artificial immune systems to deal with computer-security threats such as viruses and other closely analogous scenarios such as network intrusion detection. Focussing on the immune system's ability to extend and maintain its threat repertoire leads to the idea of using artificial immune systems in machine learning and monitoring applications. Once embroiled in the implementation of such systems however it is very easy to drift away from initially biologically inspired mechanisms, and to produce systems which pay lip-service but owe little else to the immune system.

In many ways it is hard to see why we might wish to apply the word "cognitive" to the immune system when we seek its inspiration for such comparatively lowly tasks.

Autonomy, survival and homeostasis

A key concern in robotics, especially for mobile robots and robots in harsh environments such as space is the notion of *autonomy*. This is usually defined as the ability to continue operation in the absence of outside

assistance. Clearly this is an extremely wooly definition and prone to reinterpretation whenever convenient. Many workers in robotics see autonomy and survival as essentially synonymous, although the continuation of useful behaviour is sometimes advocated as necessary for autonomy. The concept of homeostasis at an organism level is also not very clearly defined, but certainly includes consideration of internal non-behavioural factors which are important in robots for autonomy as well as organisms for survival.

I propose that the consideration of biological metaphors in the context of their contribution to the maintenance of homeostasis (or something analogous in artificial systems) is a more enlightening and often appropriate viewpoint.

This certainly applies in the case of robotics, computer security and network intrusion; but may also be suitable in other cases. I suspect that we have all suffered from the effects of computer viruses and worms and seen their debilitating effects, but I would also propose that an anti-virus checker running on an uninfected lap-top computer can be just as debilitating, it may drain the battery to the point of uselessness and "deny service" just as effectively as a virus or worm.

Conclusion

Perhaps this is just another cry for taking a "holistic" view, but I believe it is rather more than that. I propose that if we are to be biologically motivated in our approach to software systems, then we ought to see beyond the specific mechanisms and to consider the wider implications of biological motivation. Most biological processes within organisms contribute in some way to the maintenance of homeostasis and we should bear that in mind when plundering the mechanisms for unrelated purposes.

An Investigation into Immunological Robot Controllers

Andrew Webb, Emma Hart, Peter Ross, Alistair Lawson

Napier University, Scotland, UK
{e.hart, p.ross, a.webb, al.lawson}@napier.ac.uk

Abstract. Much of current robot research is about learning tasks in which the task to be achieved is pre-specified, a suitable technology for the task is chosen and the learning process is then experimentally investigated. A more interesting research question is how can robot be provided with an architecture that would enable it to developmentally 'grow-up' and accomplish complex tasks by building on basic built-in capabilities. Previous work by the authors defined the requirements of a robot architecture that would enable this to happen — in this paper, we describe how some components of such an architecture can be achieved using an immune network model, and present preliminary results that show the plausibility of the suggested approach.

1 Introduction

A great deal of current research work in mobile robotics and autonomous systems is still focused on getting a robot to learn to do some task such as pushing an object to a known location or running as fast as possible over rough ground. The learning process may be supervised, unsupervised or a process of occasional reinforcement, but the whole aim in such work is to get the robot to achieve the task that was pre-defined by the researcher.

As a step towards achieving truly autonomous robots that can function productively for long periods in unpredictable environments, it is important to investigate how one might design robots that are capable of 'growing up' through experience. By this, we mean that the robot starts with only some basic skills such as an ability to move about and an ability to sense and react to the world , but in the course of time it develops genuinely new skills that were not entirely engineered into it at the start. In particular it should be capable of building some kind of hierarchy of skills, such that for each new skill s_{new} there is one or more sets of skills $S_1, S_2, \cdots S_n$ such that s_{new} is significantly more easily acquired if the robot has acquired all the members of some S_i than if it lacks at least one member of each of those sets. To achieve this requires a fundamental shift in thinking when designing robotic architectures compared to the type of systems prevalent in the literature today.

Previous work by the authors [1] attempted to lay out a research agenda by which this question could be answered and identified six essential ingredients of

an architecture that can realise growing-up robots. These are: sensors, memory, data-abstraction, planning, motivation, and finally a developmental schedule. [1] provides an overview of existing developmental architectures in relation to the above features. In this paper, we argue that an immune-network model can form the central component of a new architecture, which in particular provides a convenient method for handling the first four requirements. The immune network model was first proposed by Jerne in [7], and suggested that antibodies not only recognise foreign antigens, but also are connected together in a large-scale network formed by chains of stimulation and suppression between communicating antibodies. Although still controversial in immunological circles, the model has been successfully adopted by many AIS practitioners, producing diverse applications from data-mining systems [16] to simple robot-control architectures [4, 10, 14].

In the next sections, we describe the proposed architecture in detail and provide results of some early experimentation. Although this in no way represents the complete architecture and is tested only in simulation, it does at least point to the plausibility of the model.

2 Previous Work

AIS ideas have already appeared in robotics research. Lee [9] proposed an AIS for realisation of cooperative strategies and group behaviour in collections of mobile robots, and Singh and Thayer [13, 15] proposed another architecture for coordination and control of large scale distributed robot teams based on concepts from the immune system. Of more relevance to this research is the work of Ishiguro and Watanabe who introduce an immune-network for behaviour-arbitration in [4, 17], for gait-control in walking robots [5] and also the work of [10] who also consider an immune network for decentralised autonomous navigation in a robot. In some senses, this work suffers from the same problems as other robotic approaches in that it results in a control module that is essentially static, i.e. successfully implements certain fixed behaviours, but would not permit a robot to 'grow-up' in the developmental sense outlined in the introduction. However, the overall approach contains many elements that can be incorporated into our proposed system and hence is briefly outlined here.

In [4, 17], antibodies are formed into a network that successfully arbitrates between simple behaviours on a real robot; initially they hand-crafted antibodies, in later work they evolved them. An antibody consists of a paratope defining a desirable condition and related motor-action, and an idiotope which identifies other antibodies to which the idiotope is connected. Connection between the idiotope of one antibody x and the paratope of an antibody y stimulates the antibody y, and links between antibodies in the network can either be evolved by a genetic algorithm [17] or formed via an on-line adaptation mechanism which provides reinforcement signal to links, [5]. The architecture which we propose must also handle behaviour arbitration, however we wish to construct it in such a way that its links also express *sequences* of actions, and thus paths in the

network represent both a past history of robot actions (i.e. an episodic memory) and also provide information useful for planning.

A related line of research to AIS is that of the application of *classifier systems* to robot-control. Rules in a classifier system consist of conditions which are matched against the current state of the environment, and associated actions which are executed by the 'winning' rule. Such systems have been used to control a robot in simulation, for example [18] and also animats navigating in environments containing aliasing states, for example [8]. However, although these system generate control rules automatically, individual rules are distinct and there is no interaction between rules, therefore a pure classifier system approach cannot represent sequences of actions which is essential if the goals of this research are to be met. However, both the work of [18] and [8] partially informs the architecture proposed here, in particular in the chosen representation of antibodies in the network with regard to representing sensor information and motor actions.

Finally, [2] proposes a developmental mechanism which has some similarities to the proposed method, but it is not clear whether his system is scalable. His work, and its relation to our proposed model, is further discussed in section 3.1.

3 A New Architecture

Let us suppose that at the very start, the robot is driven by basic instincts such as a 'desire to avoid collisions' and a 'desire to seek novelty'. The robot should learn through experience, and the learned behaviours should gradually take over control from the instinct-driven initial system. The robot therefore needs to capture some minimal details of its experiences. In the proposed model, depicted in figure 1, this information is held as a collection of *rule-like associations* (RLAs). Each RLA is a node in a network and consists of a (partial) description C of sensory information, a robot action command A and a partial description of the sensory effects E of doing the action. After creation, an RLA therefore expresses some of the expected results of doing action A in a context C, and weighted network links express the sequencing information; a sub-path involving strongly positive weights would express an episode.

In immunological terminology, antibodies correspond to these RLAs, and antigens correspond to sensory data (not necessarily just raw data, see below); the C and E parts of an RLA can be regarded as paratope and epitope. Much as in Jerne's [7] immune-network hypothesis, connections are formed and adjusted by a process of *recognition* between the paratope of one antibody and the epitope of another, and result in stimulation and suppression of one antibody by another, according to a dynamical equation of the form given in equation 1, as first suggested by Farmer in [3]. In this equation, $a_i(t) \geq 0$ represents the strength or concentration of antibody i at time t, e_i represents the stimulation of antibody i by the antigen (current sensory information), the first summation term represents the total stimulation of the antibody i from the other antibodies in the

network, the second summation term represents the suppression of antibody i from other antibodies in the network, and k_i is a natural decay factor.

$$\frac{da_i(t)}{dt} = \left(e_i + \sum_{j=1}^{N} m_{ij} a_j(t) - \sum_{j=1}^{N} m'_{ji} a_j(t) - k_i \right) a_i(t) \quad (1)$$

Immune system models require a mechanism by which recognition can occur. For example, AIS network models (e.g. [16]) often use the Euclidean distance in data-space between two data-items to signify recognition. In the proposed architecture, recognition between RLAs or antibodies serves the following purposes:

- individually, they can express a temporal association between RLAs – a strong positive connection between X and Y means that if RLA X fits the current situation then RLA Y is a possible candidate to describe the subsequent situation. Thus, individually, they can capture some aspects of episodic memory. Importantly, the boundaries of episodes can emerge from the dynamics of the network. That is, an episode ends when there is no clear winner as to the successor RLA.
- individually, they can (as inhibitory links) express a competition between different RLAs to account for the current situation.
- collectively, they can act as an attentional mechanism. The dynamics of the network can cause it to settle to a state in which some set of RLAs are reasonably active and the remainder are not; the active set represents the 'current memory context', as it were. In the set of linear differential equations in equation 1 above, the system can only have either a point attractor or a limit cycle depending on the values of the constants involved, but note that the e_i would normally be time-dependent (the external data changes with robot activity) so the system should be capable of flipping between different attractors and limit cycles as the robot moves and the environment changes.

Clearly, this approach raises several questions. By what process(es) are RLAs created, and on what time scales? How are the connections between antibodies formed, and the strength of their affinities quantified? How will the dynamics of the network operate? We consider these questions next.

3.1 Generating RLAs

Although the aim of the architecture is to provide a framework in which the robot can grow-up, it seems reasonable to start with a system that has built-in basic behaviours, for example "explore", "avoid obstacles", "avoid boredom". We propose that this is handled by a partially pre-built network of RLAs, which then undergoes adaptation and growth until it becomes capable of allowing the robot to perform non-trivial and purposeful-seeming sequences of actions.

First, there is a short fixed-length queue that contains recent interesting sensory and motor events. The queue provides a form of short-term or working memory and distantly resembles human short-term memory which experimental

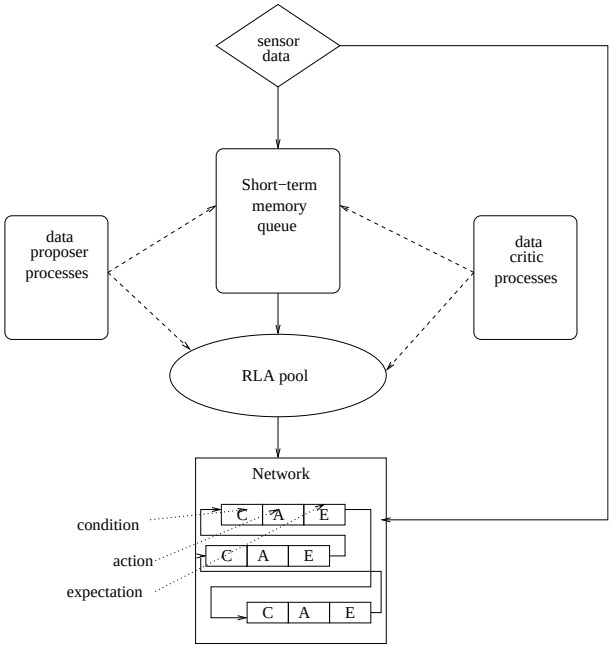

Fig. 1. A schematic representation of the proposed architecture

studies have suggested is of bounded capacity (although expandable through lengthy training) and contains things that are fairly closely linked to the sensory input. For our initial purposes, 'interesting' means 'significantly changing'; for example if the robot is moving straight ahead across a vast empty space, the queue should not alter. The contents of this queue provide the raw material from which candidate RLAs can be built and then inserted into the network. Clearly the queue needs to contain some consequences of an action before this can happen, so RLAs can only get created at certain moments. The RLA pool can be viewed as containing fragments of experience. We propose RLAs of the following form:

```
RLA-3:
   condition: front-sensors = high
              and left-sensors = low
              and moving-average-of-front-sensors = low
      action: turn right
 expectation: left-sensors = high
  activation: 0.05
       links: 7/0.9, 453/-0.2, 64/1.2
```

Note that the condition does not fully describe the raw sensor data, and may refer to higher-level data constructs at later stages of the robots development. At the

very start only raw sensor data will be available, but in real application, this will contain far too much information to be useful. So, abstractions will be proposed – for example, natural ones to suggest at the start would contain either thresholded or thresholded-moving-average versions of raw sensory information. We envisage that there will be some *data proposer and data critic processes* that suggest and evaluate new data abstractions built out of all existing data items (whether raw or already abstracted). Thus, the data universe will be dynamic. It is envisaged that the RLA proposer processes will gradually generate RLAs representing higher and higher levels of knowledge, thus representing the robot 'growing-up' in terms of its capabilities to understand its world. Thus, for example, an early set of RLAs composed of raw sensor data indicating that a robots left and front sensors are high, might eventually be replaced by an RLA representing the concept 'corner', with an associated action to turn right. Note that this thinking has some similarities with the work of Drescher [2] who introduced a general learning and concept-building mechanism called the *schema mechanism* in order to reproduce aspects of Piagetian cognitive development [11] during infancy. In this mechanism, the world is initially represented only in terms of very simple motor and sensor elements. Crucially however, the mechanism can define new, abstract, actions and invent novel concepts by constructing new state elements to describe aspects of the world that the existing repertoire of representations fails to express. Eventually, representations are discovered which can represent an object independently of how it is currently perceived and may be far removed from the original description.

Newly-formed RLAs will be presented to the network, where they will survive by being found to be useful and continue to survive only by continuing to be useful. Conversely, RLAs will be removed from the network if their stimulation falls below some threshold value. Data proposer processes are likely to be based on clustering techniques, for example k-means clustering or self-organising maps. Recent work by Prem *et al* in relation to this architecture shows promising results in using the ISO-map technique [12] for finding abstractions in time-series of sensor data generated by a real-robot. Data critic processes are likely to be based on checking whether data items have become redundant.

3.2 Quantifying Recognition between RLAs

As already stated, there is no straightforward way of quantifying the extent to which one RLA in the network should recognise another. As already mentioned in section 2, [10, 17] tackled this problem by using a genetic algorithm, but this method has significant disadvantages if the goals of the new architecture are to be achieved. Firstly, use of a GA is likely to be too computationally expensive and slow in a real robotic environment, and furthermore, the connection strengths between antibodies could possibly change over time as the robot learns more about its environment, which would require the use of a continuously running GA. This type of process does not really have an analogy in the biological immune system in which connection strengths are determined by physical binding processes which do not alter over time, but there is an obvious analogy with

the kind of Hebbian learning processes occurring in neural networks in which connection strengths are continuously adjusted over time.

However, [5] describes use an on-line adaption mechanism in an immune-network for achieving behaviour-arbitration — in this mechanism, affinity values are adaptively modified until the required behaviour emerges. This type of approach familiar to reinforcement learning appears to be more promising when using real robots, and hence will be adopted in this architecture.

3.3 Network dynamics

As mentioned in section 3.1, the RLA pool can be thought of as containing fragments of experience which may become incorporated into the network. Initially, the network should consist of instinct driven behaviours but over time, these should be replaced by more sophisticated behaviours — however, it seems reasonable that the network should still maintain some record of these instinctive behaviours, as they may be useful at points in the future, and hence can override other behaviours given the right conditions.

Biological and neurological studies tell us that the network cannot be infinitely large; the brain has a finite volume in which neurons can exist, and similarly the immune system cannot physically contain an infinite number of antibodies (and anyway, the number of different types of antibodies is limited by the diversity of the DNA from which they can be formed) hence it seems logical and practical that the size of the network must somehow be bounded. Various mechanisms for achieving this can be found in the literature; plausible ones would seem to be based on the notion of a competition for resources, where RLAs would have to prove their worth to be allowed to remain in the network else be replaced by others. The natural decay constant k_i of the antibody would aid this process but further 'cell-death' mechanisms need to be investigated.

3.4 The emergence of planning

Planning-like behaviour should emerge from the network: this could occur as a dynamic cascade of internal events. For example, a goal is represented as an antigen which is injected into the system. As in the immunological system, the network must respond to this antigen - the antigen (goal) remains in the system until it is satisfied. At any point in time, the external environment will consist of multiple and changing data items, representing goals, sensory information and (perhaps) maps and internal memory states; the resulting course of action is results from a chain of RLAs firing, determined by the dynamically changing concentrations of the antibodies. Thus, the network effectively records chains of events that can allow a desired goal to be achieved. This may lead to the emergence of more complex behaviours.

Alternatively, a more classical planning approach could be taken. The RLAs associate expectations with states, therefore in theory a *virtual* antigen could be injected into the system, representing some potential goal or action, and the dynamical equations applied to determine what would be the result of such an

action. By comparing the results of a number of such virtual experiments, a 'plan' could then be selected. The network thus provides a blackboard for 'thought' experiments by the robot.

4 Current Implementation

As a proof of concept the previous work [19] used a set of hand-crafted rules and Olivier Michel's simulation (http://diwww.epfl.ch/lami/team/michel/khep-sim/) of a Khepera robot. Though it was only a basic outline of the proposed system, the initial experiments showed that it could capture some episodes of experience. Michel's simulator has since been superceded by a purpose-built simulator that was chosen because it allows a richer sensory environment that can include colour, ultrasonic and infra red sensors. Also, different robot worlds can be created using coloured shapes. The current simulator no longer emulates the Khepera; instead, it emulates a KURT2 robot, since other collaborators on this project use the KURT2 as their robot of choice. The IR and US sensors are arranged and numbered as shown in figure 2. Sensors 0-11 are infrared, sensors 12 and 13 are ultrasonic. For KURT2 details see http://ais.gmd.de/BAR/KURT2/ in English or http://www.kurt2.de in German.

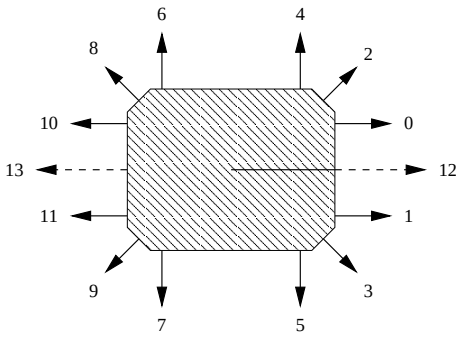

Fig. 2. The sensors

Rather than use a set of hand-crafted RLAs, as in the previous experiments, the current system can now propose new rules automatically and uses cloning and mutation mechanisms to evolve and refine them. Cloned RLAs are produced if there are no existing RLAs that sufficiently match the current sensory condition, or if the winning RLA did not receive positive reinforcement. The links connecting RLAs are still modified via an adaptation mechanism familiar to reinforcement learning, whereby connection strengths are adaptively modified to encourage the desired robot behaviour.

RLAs can also be removed from the network if their total stimulation falls below a threshold, or if they are inaccurate or no longer useful. The stimulation function currently used is a simplified version of that shown in equation 1,

whereby the RLA stimulation is a summation of the affinity with the current sensor data (antigen), total stimulation received from other RLAs and the total suppression received from other RLAs. This is shown in equation 2:

$$Stimulation = affinity + \sum stim - \sum supp \tag{2}$$

where the first and second summations represent the total stimulation and total suppression received by the neighbouring RLAs respectively. Ongoing experiments seem to suggest that the RLA network can be made to learn simple behaviours such as wall following and obstacle avoidance and can capture sensory experiences in a reasonably stable manner when run over large numbers of iterations. We are currently looking at ways of eliminating redundant RLAs from the network. At present, the system does not appear to suffer from the often-observed problem of uncontrolled expansion, though we intend to investigate Mark Neal's [16, 20] work on resource allocation, which eliminates network antibodies by sharing out some notional resource among the network cells, so that those with insufficient resource are culled from the network. It seems biologically plausible that the network cannot be of an infinite size.

Other current work is focusing on how the network adapts when the robot is placed in dynamically changing environments, i.e. can an existing network adapt quickly and efficiently during the robot's life span? We are especially interested in finding out how the network of RLAs changes or grows during the period when the robot is placed in different worlds. Related to this is how the network adapts to a changing data universe. For example the amount and type of sensory information available could be decreased or increased at various points in the simulation. Arising from this is the question of whether RLAs could be made suitably concise, rather than all of them being fully descriptive.

At present, the robot deals mainly with raw sensor data, but if it is expected to grow up and perform more sophisticated behaviours, then RLAs representing higher and higher levels of knowledge will need to be generated. This would represent part of the 'growing up' process in terms of its capability to understand the world. A simple example may be that an early RLA comprising of sensor data that indicates that the front and left sensors are high, might eventually be replaced by an RLA representing the concept of 'corner'. We are investigating how clustering and/or network algorithms could be used on segments of the RLA network to replace sequences of RLAs with new higher-level macro RLAs.

4.1 Initial Results

After running the algorithm for an initial learning phase of 100000 iterations, the RLAs chosen by the algorithm were recorded. A flow diagram showing the sequences of surviving RLAs is shown in figure 3. Note that this is not the immune network topology, but is instead used to illustrate how the different sensory experiences are captured when using the immune network approach. As an example, the RLA sequence 0, 1, 5, 6, 8, 0 captures the sequence of events that occur when the robot meets an obstacle head-on and turns to avoid it.

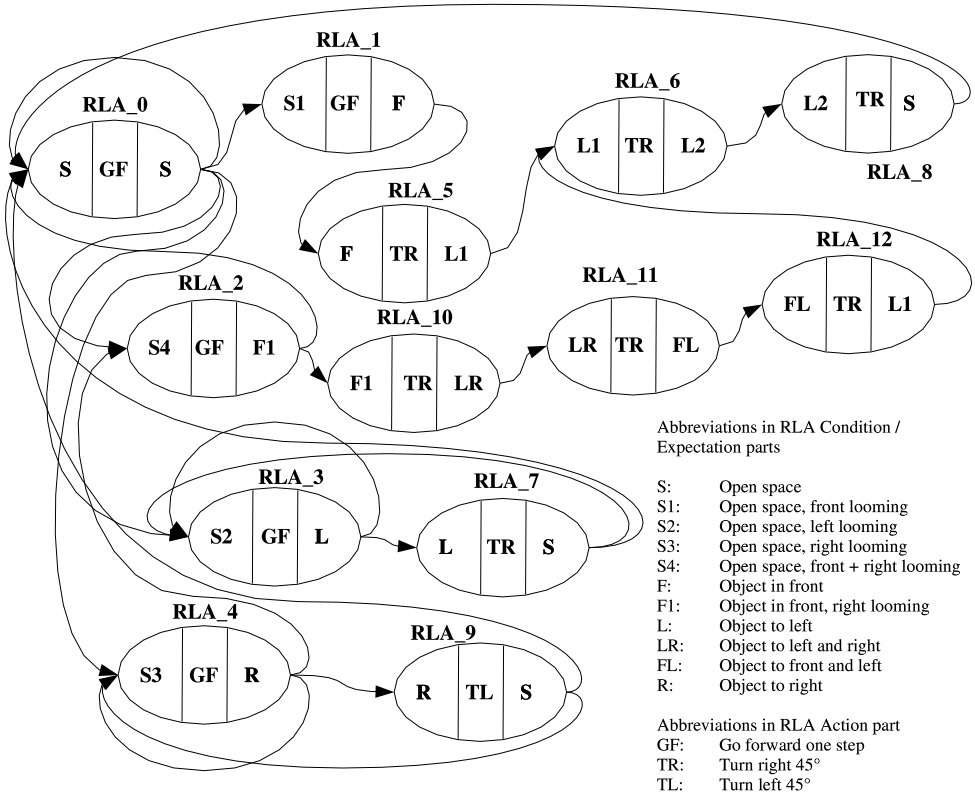

Fig. 3. Flow diagram showing sequences of chosen RLAs

This sequence could be interpreted as: 'sensing clear space, go forward, obstacle looming in front, go forward, obstacle ahead, turn right, obstacle to the left, turn right, obstacle more to the left, turn right, sensing clear space, go forward' and so on.

The remaining RLAs appear to capture some episodes (sequences of sensory events) in a reasonably stable manner, thus the robot could be said to have a long-term memory that maintains a record of the relationships between sensory situations, actions performed and the effects of those actions.

5 Conclusion

In this paper we have proposed a robot control architecture based on an AIS that should be capable of capturing at least some aspects of 'growing up' through experience. An initial experiment showed that it seemed to be capable of capturing some episodes of experience. We stress that the aim of the project is not to

see how well the immune-based system can perform specific tasks, such as wall-following or finding things, but that the real interest is in the developmental procedures that allow the robot to develop higher-level skills.

Experiments are currently being conducted in the areas of dynamically changing data universes and environments and techniques to replace sequences of RLA firings with higher-level macro RLAs. Much work also needs to be performed in investigating the scalability of the system. Furthermore, it is well known in robotics research that simulated systems rarely transfer seamlessly to the real-world, therefore we fully intend to transfer this architecture to a real-robot (see [6]).

However, we do believe that what we have sketched out represents a very fruitful line of work, both in terms of studying robot development and in terms of studying AISs. Too much research in AISs still relies on overly-simplistic metaphors. We claim that the problems of robot development provide an excellent context for studying AIS issues such as sophisticated matching algorithms, the dynamics of network models, the problems of handling a continually-evolving representation and even the computational tractability of AISs.

Acknowledgements

This work is supported by the European Union funded IST programme, grant no. IST-2000-29225.

References

1. Anonymous for reviewing purposes. Requirements for getting a robot to grow-up. In *Proceedings of the European Conference on Artificial Life, ECAL 2003*, 2003.
2. G. Drescher. *Made-Up Minds: A Constructivist Approach to Artificial Intelligence*. MIT Press, 1991.
3. J.D Farmer, H Packard N, and A.H Perelson. The immune system, adaption and machine learning. *Physica D*, 22:187–204, 1986.
4. A. Ishiguro, T. Kondo, Y. Watanabe, Y. Shirai, and H. Uchikawa. Immunoid: a robot with a decentralized consensus-making mechanism based on the immune system. In *Proceedings of ICMAS Workshop on Immunity-based Systems*, pages 82–92, 1996.
5. A. Ishiguro, S. Kuboshiki, S. Ichikawa, and Y. Uchikawa. Gait-control of hexapod walking robots using mutual coupled immune networks. *Advanced Robotics*, 10(2):179–196, 1996.
6. IST-2000-29225. Systemic intelligence for growing up artefacts that live. http://www.ist-signal.org.
7. N.K Jerne. The immune system. *Scientific American*, 229(1):52:60, 1973.
8. Pier Luca Lanzi. An analysis of the memory mechanism of XCSM. In *Genetic Programming 1998: Proceedings of the Third Annual Confere nce*, pages 643–65. Morgan Kaufman, 1998.
9. D. Lee, H. Jun, and K. Sim. Artifical immune system for realisation of cooperative strategies and group behaviour in collective autonomous mobile robots. In *Proceedings of Fourth Internation Symposium on Artifical Life and Robotics*, pages 232–235, 1999.

10. R. Michelan and F.J. von Zuben. Decentralized control system for autonomous navigation based on an evolved artificial immune system. In C.L. Giles et al, editor, *Proceedings of CEC-02*, pages 1021–1026. IEEE Press, 2002.
11. J. Piaget. *The Origins of Intelligence in Children*. Norton, N.Y., 1952.
12. E. Prem and P. Poelz. Concept acquistion using isomap on sensorimotor experiences of a mobile robot. In *Proceedings of International Workshop on Epigenetic Robotics*, Boston, August 2003. to appear.
13. Surya Singh and Scott Thayer. Immunology directed methods for distributed robotics: A novel, immunity-based architecture for robust control and coordination. In *Proceedings of SPIE: Mobile Robots XVI*, volume 4573, November 2001.
14. Surya Singh and Scott Thayer. A foundation for kilorobotic exploration. In *Proceedings of the Congress on Evolutionary Computation at the 2002 IEEE World Congress on Computational Intelligence*, May 2002.
15. Surya Singh and Scott Thayer. Kilorobot search and rescue using an immunologically inspired approach. In *Distributed Autonomous Robotic Systems*, volume 5. Springer-Verlag, June 2002.
16. Jon Timmis and Mark Neal. A resource limited artificial immune system for data analysis. *Knowledge Based Systems*, 14(3-4):121–130, June 2001.
17. Y. Watanabe, A. Ishiguro, Y. Shiraio, and Y. Uchikawa. Emergent construction of behaviour arbitration mechanism based on the immune system. *Advanced Robotics*, 12(3):227–242, 1998.
18. A. Webb, E. Hart, P. Ross, and A. Lawson. Controlling a simulated khepera with an xcs classifier system. In *Proceedings of 7th European Conference on A-Life*. Springer-Verlag, 2003. to appear.
19. E. Hart, P. Ross, A. Webb, and A. Lawson. A Role for Immunology in "Next Generation" Robot Controllers In *Proceedings of 2nd International Conference on Artificial Immune Systems*, pages 46–56. Springer-Verlag, 2003.
20. M. Neal. Meta-stable memory in an artificial immune network. In *Proceedings of 2nd International Conference on Artificial Immune Systems*, pages 168–180. Springer-Verlag, 2003.

Artificial Immune Recognition System (AIRS): Revisions and Refinements

Andrew Watkins

Computing Laboratory
University of Kent at Canterbury, UK
and
Department of Computer Science
Mississippi State University, USA.
abw5@ukc.ac.uk

Jon Timmis

Computing Laboratory
University of Kent at Canterbury, UK
J.Timmis@ukc.ac.uk

Abstract

This paper revisits the Artificial Immmune Recognition System (AIRS) that has been developed as an immune-inspired supervised learning algorithm. Certain unnecessary complications of the original algorithm are discussed and means of overcomming these complexities are proposed. Experimental evidence is presented to support these revisions which do not sacrifice the accuracy of the original algorihtm but, rather, maintain accuracy whilst increasing the simplicity and data reduction capabilities of AIRS.

1 INTRODUCTION

Recently, there has been a great deal of interest in the use of the immune system as inspiration for computer science and engineering. These Artificial Immune Systems (AIS) seem to have great potential, which is as yet unrealized. An intuitive application of AIS is in the area of computer security, network intrusion detection (Forrest, Perelson et al. 1994), (Hofmeyr and Forrest 2000) and (Kim and Bentley 2001), change detection, and so on. However, AIS are not limited to this field alone. Work has identified that the immune system contains certain properties that may be useful to create learning algorithms for computer science through the exploitation of the natural learning mechanisms contained within the immune system (Bersini and Varela 1990). However, the focus of current AIS research seems to have been on the development of unsupervised learning algorithms (De Castro and Von Zuben 2000b) and (Timmis and Neal 2001) rather than the supervised or reinforcement kind. An exception to this is work in (Carter 2000). Recent work in (Watkins 2001) explored the possibility of utilizing the immune system as inspiration for the creation of a supervised learning technique. By extracting useful metaphors from the immune system and building on previous immune inspired unsupervised learning algorithms, a classifier was constructed that seems to perform reasonably well on various classification and machine learning problems (Watkins and Boggess 2002a).

This paper presents a further investigation into the work of (Watkins 2001) and suggests improvements to the 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 paper outlines the previous work undertaken in (Watkins 2001), suggests improvements to the algorithms and discusses the implications of these new results. Attention is then given to future possibilities with this approach.

2 BACKGROUND RESEARCH ON AIRS

AIRS (Artificial Immune Recognition System) is a novel immune inspired supervised learning algorithm (Watkins 2001). Motivation for this work came from the author's identification of the fact that there was a significant lack of research that explored the use of the immune system metaphor for supervised learning; indeed, the only work identified was that of (Carter 2000). However, it was noted that within the AIS community there had been a number of investigations on exploiting immune mechanisms for unsupervised learning (that is, where the class of data is unknown a-priori) (Timmis, Neal et al. 2000), (Timmis and Neal 2001) and (De Castro and Von Zuben 2000b). Work in (De Castro and Von Zuben 2000a) examined the role of the clonal selection process within the immune system (Burnet 1959) and went on to develop an unsupervised learning known as CLONALG. This work was extended by employing the metaphor of the immune network theory (Jerne 1974) and then applied to data clustering. This led to the development of the aiNet algorithm (De Castro and Von Zuben 2000b). Experimentation with the aiNet algorithm revealed that evolved *artificial immune networks*, when combined with

traditional statistical analysis tools, were very effective at extracting interesting and useful clusters from data sets. aiNet was further extended to multimodal optimization tasks (De Castro and Timmis 2002b). Other work in (Timmis, Neal et al. 2000) also utilized the immune network theory metaphor for unsupervised learning, and then augmented the work with the development of a resource limited artificial immune network (Timmis and Neal 2001), which reported good benchmark results for cluster extraction and exploration with *artificial immune networks*. Indeed, this work has been further extended by (Nasaroui, Gonzalez et al. 2002) with the introduction of fuzzy logic and refinement of various calculations. The work in (Timmis and Neal 2001) was of particular relevance to (Watkins 2001) and the further work described in this paper.

Building on this previous work, in particular the ideas of artificial recognition balls and resource limitation from (Timmis and Neal 2001) and long-lived memory cells from (De Castro and Von Zuben 2000b). AIRS demonstrated itself to be an effective classifier. The rest of this section describes the immune metaphors that have been employed within AIRS, outlines the algorithm and discusses results obtained, before progressing to the following section, which describes augmentations and improvements to AIRS.

2.1 IMMUNE PRINCIPLES EMPLOYED

A little time should be taken to draw attention to the most relevant aspects of immunology that have been utilized as inspiration for this work. A more detailed overview of the immune system and its relationship with computer science and engineering can be found in (De Castro and Timmis 2002a).

Throughout a person's lifetime, the body is exposed to a huge variety of pathogenic (potentially harmful) material. The immune system contains lymphocyte cells known as B- and T-cells, each of which has a unique type of molecular receptor (location in a shape space). Receptors in this shape space allow for the binding of the pathogenic material (antigens), with the higher affinity (complementarity) between the receptor and antigen indicating a stronger bind. Work in (De Castro and Timmis 2002a) adopted the term shape-space to describe the shape of the data being used, and defined a number of affinity measures, such as Euclidean distance, which can be used to determine the interaction between elements in the AIS. Within AIRS (and most AIS techniques) the idea of antigen/antibody binding is employed and is known as *antigenic presentation*. When dealing with learning algorithms, this is used to implement the idea of matching between training data (antigens) and potential solutions (B-Cells). Work in (Timmis and Neal 2001) employed the idea of an *artificial recognition ball (ARB),* which was inspired by work in (Farmer, Packard et al. 1986) describing antigenic interaction within an immune network. Simply put, an ARB can be thought to represent a number of identical B-Cells and is a mechanism employed to reduce duplication and dictate survival within the population.

Once the affinity between a B-Cell and an antigen has been determined, the B-Cell involved transforms into a plasma cell and experiences *clonal expansion*. During the process of clonal expansion, the B-Cell undergoes rapid proliferation (cloning) in proportion to how well it matches the antigen. This response is antigen specific. These clones then go through *affinity maturation*, where some undertake somatic hypermutation (mutation here is inversely proportional to antigenic affinity) and eventually will go through a selection process through which a given cell may become a memory cell. These memory cells are retained to allow for a faster response to the same, or similar, antigen should the host become re-infected This faster response rate is known as the secondary immune response. Within AIRS, the idea of clonal expansion and affinity maturation are employed to encourage the generation of potential memory cells. These memory cells are later used for classification.

Drawing on work from (Timmis and Neal 2001), AIRS utilized the idea of a stimulation level for an ARB, which, again, was derived from the equations for an immune network described in (Farmer, Packard et al. 1986). Although AIRS was inspired by this work on immune networks, it was found that maintaining a network representation—with connections, stimulation, and repression among the ARBs in the system—was not necessary for evolving a useful classifier. In AIRS, ARBs experience a form of clonal expansion after being presented with training data (analogous to antigens); details on this process are provided in section 2.2. However, AIRS did not take into account the affinity proportional mutation. When new ARBs were created, they were subjected to a process of random mutation with a certain probability and were then incorporated into the memory set of cells should their affinity have met certain criteria. Within the AIRS system, ARBs competed for survival based on the idea of a resource limited system (Timmis and Neal 2001). A predefined number of resources existed, for which ARBs competed based on their stimulation level: the higher the stimulation value of an ARB the more resources it could claim. ARBs that could not successfully compete for resources were removed from the system. The term *metadynamics* of the immune system refers to the constant changing of the B-Cell population through cell proliferation and death. This was present in AIRS with the continual production and removal of ARBs from the population. Table 1 summarizes the mapping between the immune system and AIRS.

Table 1: Mapping between the Immune System and AIRS

IMMUNE SYSTEM	AIRS
Antibody	Feature vector
Recognition Ball	Combination of feature vector and vector class
Shape-Space	The possible values of the data vector
Clonal Expansion	Reproduction of ARBs that are well matched with antigens
Antigens	Training data
Affinity Maturation	Random mutation of ARB and removal of lowest stimulated ARBs
Immune Memory	Memory set of mutated ARBs
Metadynamics	Continual removal and creation of ARBs and Memory Cells

2.2 THE AIRS ALGORITHM

The previous section outlined the metaphors that were employed in the development of AIRS. This section now presents the actual algorithm and discusses the results obtained from experimentation. A more detailed description of the algorithm and results can be found in (Watkins 2001).

Within AIRS, each element (ARB) corresponds to a vector of n dimensions and a class to which the data belongs. Additionally, each ARB has an associated stimulation level as defined in equation 1, where x is feature vector of the ARB, s^x is the stimulation of an ARB x, y is the training antigen, and affinity, in the current implementation, is a function that calculates the Euclidean distance:

$$s^x = \begin{cases} 1\text{-affinity}(x,y), & \text{if class of } x \equiv \text{class of } y \\ \text{affinity}(x,y), & \text{otherwise} \end{cases} \quad (1)$$

Notionally, AIRS has four stages to learning: initialization, memory cell identification, resource competition and finally refinement of established memory cells. AIRS is a one-shot learning algorithm; therefore, the process described below is run for each antigenic pattern, one at a time. Each of these processes will be outlined with the algorithm summarized below.

Initialization of the system includes data pre-processing (normalization) and seeding of the system with randomly chosen data vectors. Assuming a normalized input training data set (antigens), data from that set are randomly selected to form the initial ARB population **P** and memory cells **M**. Prior to this selection, an affinity threshold is calculated; this threshold for the current implementation is the average Euclidean distance between each item in the training data set. This is then used to control the quality of the memory cells maintained as classifier cells in the system.

AIRS maintains a population of memory cells **M** for each class of antigen, which, upon termination of the algorithm, should have identified suitable memory cells to provide a generalized representation for each class of antigenic pattern. The first stage of the algorithm is to determine the affinity of memory cells to each antigen of that class. Then the highest affinity cells are selected for cloning to produce a set of ARBs (which will ultimately be used to create an established memory set). The number of clones that are produced is in proportion to the antigenic affinity, i.e., how well they match; the ARBs also undergo a random mutation to introduce diversification.

The next stage is to identify the strongest, based on affinity to the training instance, ARBs; these will be used to create the established memory set used for classification. This is achieved via a resource allocation mechanism, taken from (Timmis and Neal 2001), where ARBs are allocated a number of resources based on their normalized stimulation levels. At this point, it is worth noting that the stimulation level of an ARB is calculated not only from the antigenic match, but also the class of the ARB. This, in effect, provides reinforcement for ARBs that are of the same class as the antigenic pattern being learnt and that match the antigenic pattern well, in addition to providing reinforcement for those that do not fall into that class and do not match the pattern well.

Once the stimulation of an ARB has been calculated, the ARB is allowed to produce clones (which undergo mutation). The termination condition is then tested to discover if the ARBs are stimulated enough for training to cease on this antigenic pattern. This is defined by taking the average stimulation for the ARBs of each class, and if each of these averages falls above a pre-defined threshold, training ceases for that pattern. This ARB production is repeated until the stopping criteria are met. Once the criteria have been met, then the candidate memory cell can be selected.

A candidate memory cell is selected from the set of ARBs based on its stimulation level and class, with the most stimulated ARB of the same class as the antigen being selected as the candidate. If this candidate cell has a higher stimulation than any memory cell for that class in the established memory set **M**, then it is added to **M**. Additionally, if the affinity of this candidate memory cell with the previous best memory cell is below the affinity threshold, then this established memory cell is removed from the population and replaced by the newly evolved memory cell, thus achieving population control.

This process is then repeated for all antigenic patterns. Once learning has completed, the set of established memory cells **M** can be used for classification. The algorithm is presented below, in terms of immune processes employed.

1. *Initialization*: Create a random base called the memory pool (**M**) and the ARB pool (**P**).
2. *Antigenic Presentation*: for each antigenic pattern do:

   a) *Clonal Expansion*:

   For each element of **M** determine their affinity to the antigenic pattern, which resides in the same class. Select highest affinity memory cell (*mc*) and clone *mc* in proportion to its antigenic affinity to add to the set of ARBs (**P**)

   b) *Affinity Maturation*:

   Mutate each ARB descendant of this highest affinity *mc*. Place each mutated ARB into **P**.

   c) *Metadynamics of ARBs*:

   Process each ARB through the resource allocation mechanism. This will result in some ARB death, and ultimately controls the population. Calculate the average stimulation for each ARB, and check for termination condition.

   d) *Clonal Expansion and Affinity Maturation*:

   Clone and mutate a randomly selected subset of the ARBs left in **P** based in proportion to their stimulation level.

   e) *Cycle*:

   While the average stimulation value of each ARB class group is less than a given stimulation threshold repeat from step 2.c.

   f) *Metadynamics of Memory Cells*:

   Select the highest affinity ARB of the same class as the antigen from the last antigenic interaction. If the affinity of this ARB with the antigenic pattern is better than that of the previously identified best memory cell *mc* then add the candidate (*mc-candidate*) to memory set **M**. Additionally, if the affinity of *mc* and *mc-candidate* is below the affinity threshold, then remove *mc* from **M**.
3. *Cycle*. Repeat step 2 until all antigenic patterns have been presented.

2.3 RESULTS AND DISCUSSION

AIRS was tested on a number of benchmark data sets in order to assess the classification performance. This section will briefly highlight those results and discuss potential improvements for the algorithm, more details can be found in (Watkins and Boggess 2002a).

Once a set of memory cells has been developed, the resultant cells can be used for classification. This is done through a k-nearest neighbor approach. Experiments were undertaken using a simple linearly separable data set, where classification accuracy of 98% was achieved using a k-value of 3. This seemed to bode well, and further experiments were undertaken using the Fisher Iris data set, Pima diabetes data, Ionosphere data and the Sonar data set, all obtained from the repository at the University of California at Irvine (Blake and Merz 1998). Table 2 shows the performance of AIRS on these data sets, a full comparison table of AIRS and other techniques can be found in (Watkins and Boggess 2002a).

Table 2: AIRS Classification Results on Benchmark Data

IRIS	IONOSPHERE	DIABETES	SONAR
96.7	94.9	74.1	84.0

These results were obtained from averaging multiple runs of AIRS, typically consisting of three, or more, runs and five-way, or greater, cross validation. More specifically, for the Iris data set a five-fold cross validation scheme was employed with each result representing an average of three runs across these five divisions. To remain comparable to other experiments reported in the literature, the division between training and test sets of the Ionosphere data set as detailed in (Blake and Merz 1998) was maintained. However, the results reported here still represent an average of three runs. For the Diabetes data set a ten-fold cross validation scheme was used, again with each of the 10 testing sets being disjoint from the others and results were averaged over three runs across these data sets. Finally, the Sonar data set utilized the thirteen-way cross validation suggested in the literature (Blake and Merz 1998) and was averaged over ten runs to allow for more direct comparisons with other experiments reported in the literature. During the experimentation, it was noted by the authors that varying system parameters such as number of seed cells varied performance on certain data sets, however, varying system resources (i.e., the numbers of resources an ARB could compete for) seemed to have little affect. A comparison was made between the performance of AIRS and other benchmark techniques, where AIRS seemed not to outperform specialist techniques, but on more general purpose algorithms, such as C4.5, it did outperform.

Even though initial results from AIRS did look promising, it can be said there are a number of potential areas for simplification and improvement. There is clearly a need to understand exactly why and how AIRS behaves the way it does. This can be achieved through a rigorous analysis of the algorithm, examining the behavior of the ARB pool and memory set over time. To date, the focus has been primarily on the classification performance of AIRS. Indeed, the final chapter of (Watkins 2001) suggests that an investigation into the resource allocation mechanism would be a useful area of investigation. The majority of AIS techniques use the metaphor of somatic hypermutation or affinity proportional mutation. To date, AIRS does not employ this metaphor but instead uses a naïve random generation of mutations.

The remaining sections of this paper undertake these investigations and present a modified version of AIRS, which is more efficient in terms of ARB production, employs affinity proportional mutation and assess what, if any, difference this has made to the overall algorithm.

3 A MORE EFFICIENT AIRS

Motivated by the observations in (Watkins 2001), current work has focused on refining AIRS. This section details the observations that have been made through a thorough investigation into AIRS and how issues raised through these observations have been overcome.

3.1 OBSERVATIONS

3.1.1 The ARB Pool

A very crude visualization[1] was used to gain a better understanding of the development of the ARB pool. In AIRS there are 2 independent pools of cells, the memory cell pool and the ARB pool. The initial formulation of AIRS uses the ARB pool to evolve a candidate memory cell of the same class as the training antigen, which can potentially enter the memory cell pool. During this evolution, ARBs of a different class than the training antigen were also maintained in the ARB pool. The stimulation of an ARB was based both on affinity to the antigen and class, where highly stimulated ARBs were those of the same class as the antigen and that were "close" to the antigen, or of a different class and "far" from the antigen. However, the visualization revealed that during the process of evolving a candidate memory cell, there seems no need to maintain or evolve ARBs that are a different class than the training antigen. The point of the interaction of the ARB pool with the antigenic material is really only in evolving a good potential memory cell, and this potential memory cell **must** be of the same class as the training antigen. When observing the visualization for a while, it is possible to notice that there is a process of convergence by ARBs of the same class to the training antigen. Naturally, based on the reward scheme, ARBs of a different class are moving further away from the training antigen. However, this process essentially must start over for the introduction of each new antigen, and, therefore, previously existing ARBs are fairly irrelevant. Since there are 2 separate cell pools, with the true memory of the system only being maintained in the Memory Cell pool, maintaining any type of memory in the ARB pool is unnecessary. This change to the algorithm rather than being about resource allocation schemes as initially suggested in (Watkins 2001) is really a simplification to the algorithm, which is seen as a positive step. This simplification affects both memory usage and computational simplification, although this will not be discussed in this paper.

[1] See http://www.cs.ukc.ac.uk/people/rpg/abw5/ARB_hundred.html

3.1.2 Mutation of Cells

Motivated by observing the success of other AIS work, as well as by some of the tendencies discussed in (Watkins 2001) and (Watkins and Boggess 2002b), attention was paid to the way in which mutation occured within AIRS. In these two works, the authors notice that some of the evolved memory cells do not seem as high-quality of classifier cells as some of the others. Additionally, it was observed that there seemed to be some redundancy in the memory cells that were produced. In (De Castro and Von Zuben 2000a) and other AIS work, mutation within an antibody or B-Cell is based on its affinity, with higher affinity cells being mutated less than lower affinity cells. These other AIS works have used this method of somatic hypermutation to a good degree of success. It was thought that embedding some of this approach in AIRS might result in higher quality, less redundant, memory cells. This approach was therefore adopted within AIRS.

3.2 AIRS: WHAT IS NEW?

For the remainder of this section changes that have been made to the AIRS algorithm are described. There then follows empirical results from the new formulation and discuss the implications of these results.

3.2.1 Memory Cell Evolution

In the newly formulated version of AIRS, candidate memory cell evolution is based only on ARBs of the same class as the training antigen. This means that ARBs in the ARB pool are no longer permitted to mutate class. Therefore, the ARB pool will only consist of ARBs that are of the same class as the training antigen. At the end of each antigenic presentation cycle, the pool can be either be cleared out, or the ARBs can stay in the pool. If the pool is not cleared out then it will contain ARBs of all potential classes. The algorithm is only reinforcing the class of the antigenic pattern, and therefore, all ARBs that are in the pool at the end of the antigenic cycle that are not of the same class as the antigenic pattern will be removed through the metadynamic process, as they are no longer rewarded with any resources. This is in contrast to the original formulation of AIRS in which the

allocation of resources, and thus cellular reinforcement, was based on a stimulation value that was calculated as in Equation 1 (section 2.2). In that original version both ARBs "near" the antigen and of the same class as the antigen were rewarded and ARBS "far" from the antigen and of a different class than the antigen were rewarded. Also, ARBs were allowed to mutate their class values (mutate in this case means switching classes). In the newly proposed version of AIRS, only ARBs of the same class are rewarded and mutation of the class value is no longer permitted.

Based on this new formulation, the only user parameter changes that might need to be made is that the stimulation threshold could potentially need to be raised. Recall, that the stimulation threshold was used as a stopping criterion for training the ARB pool on an antigen. In order to stop training on an antigen the average normalized stimulation level had to exceed the stimulation threshold for each class group of ARBs. That is, in a 2-class problem, for example, the average normalized stimulation level of all class 0 ARBs had to be above the stimulation threshold, and the average normalized stimulation level of all class 1 ARBs has to be above the stimulation threshold. It was possible, and frequently the case in fact, that the average normalized stimulation level for the ARBs of the same class as the training antigen reached the stimulation threshold before the average normalized stimulation level of ARBs in different classes from the antigen. What this did, in effect, was allow for the evolution of even higher stimulated ARBs of the same class while they were waiting for the other classes to reach the stimulation threshold. By taking out these extra cycles of evolution through no longer worrying with ARBs of different classes, it is possible that the ARBs will not have converged "as much" as in the previous formulation. This can be overcome by raising the stimulation threshold and thus requiring a greater level of convergence.

3.2.2 Somatic Hypermutation

To explore the role of mutation on the quality of the memory cells evolved, the mutation routine was modified so that the amount of mutation allowed by a given gene in a given cell is dictated by its stimulation value. Specifically, the higher the normalized stimulation value, the smaller the range of mutation allowed. Essentially, the range of mutation for a given gene = 1.0 - the normalized stimulation value of the cell. Mutation is then controlled over this range with the original gene value being placed at the center of the range. This, in a sense, allows for tight exploration of the space around high quality cells, but allows lower quality cells more freedom to explore widely. In this way, both local refinement and diversification through exploration are achieved.

3.3 THE AIRS V2 ALGORITHM

The changes made to the AIRS algorithm are small, but end up having an interesting impact on both the simplicity of implementation and on the quality of results. Section 4 will offer more discussion by way of comparison. For now, the changes to the original AIRS presented in section 2.2 will be discussed. These can be identified as follows:

1. Only the Memory Cell pool is seeded during initialization rather than both the MC pool (**M**) and the ARB pool (**P**). Since we are no longer concerned about maintaining memory or class diversity within **P** it is no longer necessary to initialize **P** from the training data or from examples of multiple classes.
2. During the clonal expansion from the matching memory cell used to populate **P**, the newly created ARBs are no longer allowed to mutate class. Again, maintaining class diversity in **P** is not necessary.
3. Resources are only allocated to ARBs of the same class as the antigen and are allocated in proportion to the inverse of an ARB's affinity to the antigen.
4. During affinity maturation (mutation), a cell's stimulation level is taken into account. Each individual gene is only allowed to change over a finite range. This range is centered with the gene's pre-mutation value and has a width the size of the difference of 1.0 and the cell's stimulation value. In this way the mutated offspring of highly stimulated cells (those whose stimulation value is closer to 1.0) are only allowed to explore a very tight neighborhood around the original cell, while less stimulated cells are allowed a wider range of exploration. (It should be noted that during initialization all gene values are normalized so that the Euclidean distance between any two cells is always within one. During this normalization, the values to transform a given gene to within the range of 0 and 1 are discovered, as well. This allows for this new mutation routine to take place in a normalized space where each gene is in the range of 0 and 1.)
5. The training stopping criterion no longer takes into account the stimulation value of ARBs in all classes, but now only accounts for the stimulation value of the ARBs of the same class as the antigen. In the new formulation of AIRS it is still possible to have ARBs in **P** of different classes if the implementation does not clear the ARB pool after each antigenic pattern. However, this will not affect the stopping criterion since the changes to the algorithm now only require that the average stimulation value of the ARBs of the **same** class as the antigen be above the user-supplied stimulation threshold.

3.4 RESULTS AND DISCUSSION

To allow for comparison between the two versions of the algorithm, the same experiments were performed on the new formulation of AIRS (AIRS2). Section 4 will provide a more thorough comparative discussion, but for now, results of AIRS2 on the four, previously discussed, benchmark sets are presented in Table 3.

Table 3: AIRS2 Classification Results on Benchmark Data

IRIS	IONOSPHERE	DIABETES	SONAR
96.0	95.6	74.2	84.9

These results were obtained by following the same methodology as the original results reported in section 2.3 which is elaborated upon in (Watkins 2001) and (Watkins and Boggess 2002a). Again, we note that these results are competitive with other classification techniques discussed in the literature, such as C4.5, CART, and Multi-Layer Perceptrons.

4 COMPARATIVE ANALYSIS

This section briefly touches on some comparisons between the original version of AIRS presented in discussed in section 2 (AIRS1) and the revisions to this algorithm presented in section 3 (AIRS2). The focus of this discussion will be on two of the more important features of the AIRS algorithms: classification accuracy and data reduction.

4.1 CLASSIFICATION ACCURACY

The success of AIRS1 as a classifier (cf, (Watkins and Boggess 2002a)) makes it important to assess any potential changes to the algorithm in light of test set classification accuracy. To aid in this task, Table 4 presents the best average test set accuracies, along with the standard deviations, achieved by both versions of AIRS on the four benchmark data sets.

Table 4: Comparative Average Test Set Accuracies

	AIRS1: Accuracy	AIRS2: Accuracy
Iris	96.7 (3.1)	96.0 (1.9)
Ionosphere	94.9 (0.8)	95.6 (1.7)
Diabetes	74.1 (4.4)	74.2 (4.4)
Sonar	84.0 (9.6)	84.9 (9.1)

It can be noted that the revisions to AIRS presented in section 3 do not require a sacrifice in classification performance of the system. In fact, for 3 of the 4 data sets we see a slight improvement in the accuracy; however, these differences are not statistically significant. What is important to note is that the changes introduce no fundamental differences in classification accuracy of the system.

4.2 DATA REDUCTION

From the previous subsection it can be seen that the changes introduced to AIRS offer no real difference in classification accuracy, so the question arises: why bother? Why introduce these changes to a perfectly reasonably performing classification algorithm? The answer lies in the data reduction capabilities of AIRS.

In (Watkins 2001) and (Watkins and Boggess 2002b), the authors discuss that aside from competitive accuracies another intriguing feature of the AIRS classification system is its ability to reduce the number of data points needed to characterize a given class of data from the original training data to the evolved set of memory cells. Given the volumes of data involved with many real-world data sets of interest, any technique that can reduce this volume while retaining the salient features of the data set is useful. Additionally, it is this collection of memory cells that are the primary classifying agents in the evolved system. Since classification is, currently, performed in a *k*-nearest neighbor approach, whose classification time is dependent upon the number of data points used for classifying a previously unseen data item, any reduction in the overall number of evolved memory cells is also useful for the algorithm.

Table 5 presents the average size of the evolved set of memory cells and the amount of data reduction this represents in terms of population size and percentage reduction, along with standard deviations, for each version of the algorithm on the four benchmark data sets. The original training set size is also presented for comparison. There are two points of interest:

1. Both versions of the algorithm exhibit data reduction, and
2. AIRS2 tends to exhibit greater data reduction than AIRS1.

Table 5: Comparison of the Average Size of the Evolved Memory Cell Pool

	Training Set Size	AIRS1: Memory Cells	AIRS2: Memory Cells
Iris	120	42.1/65% (3.0)	30.9/74% (4.1)
Ionosphere	200	140.7/30% (8.1)	96.3/52% (5.5)
Diabetes	691	470.4/32% (9.1)	273.4/60% (20.0)
Sonar	192	144.6/25% (3.7)	177.7/7% (4.5)

		(3.7)	(4.5)

This second point is the more important for our current discussion. As mentioned in sections 3.1.2 and 3.2.2, one of the goals of the revision of the AIRS algorithm was to see if employing somatic hypermutation through a method more in keeping with other research in the AIS field would increase the efficiency of the algorithm. The current measure of efficiency under concern is the amount of data needed to represent the original training set to achieve accurate classifications. We can see from Table 5 that, in general, AIRS2 was able to achieve the comparable accuracy presented in section 4.1 with greater efficiency. In fact for some of the data sets, most notably Ionosphere and Diabetes, the degree of data reduction is greatly increased (from 30% to 52% for Ionosphere data and from 32% to 60% for the diabetes data set). Interestingly, for the most difficult classification task, the Sonar data set, the degree of data reduction is not increased. While this was not the general trend on this data set (data not presented), it does possibly point to some limitations in the current version of AIRS. Overall, however, it seems reasonable to claim that the revisions to AIRS provide greater data reduction, and hence greater efficiency, without sacrificing accuracy.

4.3 A WORD ABOUT SIMPLICITY

While the focus has not been on algorithmic complexity analysis of the two versions of AIRS for this current paper, it would be remiss not to make a brief mention concerning the simplifying effects of the revision to AIRS. As mentioned in section 3.1, the reformulation of AIRS was chiefly motivated by some basic observations about the workings of the system. One observation was that the original version of AIRS maintained representation of too many cells for its required task. This led to the elimination of maintaining multiple classes of cells in the ARB pool or of retaining cells in the ARB pool at all. This has the simplifying effect of reducing the memory necessary to run the system successfully. A second observation concerning the quality of the evolved memory cells led to the investigation of the mutation mechanisms employed in the original algorithm. By adopting an approach to mutation proven to be successful in other AIS, it has been possible to increase the quality of the evolved memory cells that is evidenced by the increased data reduction without a decrease in classification accuracy. Both of these overarching changes (ARB pool representation and the mutation mechanisms used) have exhibited a simplifying effect on the classification system as a whole.

5 CONCLUSIONS AND FUTURE WORK

This paper has focused on a supervised learning system based on immunological principles. The Artificial Immune Recognition System (AIRS) introduced in (Watkins 2001) exhibited initial success as a classification algorithm. However, as with any initial system, there were some revisions and refinements that could be made to AIRS that would decrease the complexity of the system. This paper has presented investigations for two of these revisions.

It was shown that the internal data representation of the original version of AIRS was overcomplicated. By simplifying the evolutionary process, it was possible to decrease this complexity whilst still maintaining accuracy. It was also shown that the use of affinity aware mechanisms of somatic hypermutation, as adopted throughout the AIS community, led to higher quality memory cells in AIRS and thus greater data reduction and faster classification of test data items.

Both of these revisions were the result of careful observation of the behavior of the original algorithm. In this respect, it can be said that this paper is also about the importance of taking the steps to investigate the behavior of a system even if it is performing in a successful manner. This paper has demonstrated that such an investigation is fruitful in simplifying the workings without sacrificing the performance of the system.

There are many avenues that can be explored with this work. One is the analogy of this work with reinforcement learning strategies, it could possibly be argued that AIRS is a reinforcement learning algorithm, when one considers certain mechanism within the immune system (Bersini and Varela 1994); this warrants further investigation. Additionally, the role of parallel and distributed processing could be examined, in order to allow for dealing with larger scale problems. Work has already begun on applying AIRS to immunological data, attempting to predict the binding of receptors and in effect trying to solve an immunological problem with an artificial immune system.

References

Bersini, H. and F. J. Varela (1990). Hints for Adaptive Problem Solving Gleaned from Immune Networks. *Parallel Problem Solving from Nature*. pp.343-355

Bersini, H. and F. J. Varela (1994). The Immune Learning Mechanisms: Reinforcement, Recruitment and their Applications. *Computing with Biological Metaphors*. R. Paton, Chapman and Hall**:** 166-192.

Blake, C. L. and C. J. Merz (1998). UCI Repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html

Burnet, F. M. (1959). The Clonal Selection Theory of Immunity, Vanderbilt University Press, Nashville, TN.

Carter, J. H. (2000). "The Immune System as a model for Pattern Recognition and Classification." *Journal of the American medical Informatics Association* **7**(1).

De Castro, L. N. and J. Timmis (2002b). "An Artificial Immune Network for Multimodal Optimisation." *Congress on Evolutionary Computation. Part of the World Congress on Computational Intelligence:* 699-704.

De Castro, L. N. and J. I. Timmis (2002a). Artificial Immune Systems: *A New Computational Intelligence Approach*, Springer-Verlag.

De Castro, L. N. and F. Von Zuben (2000a). "The clonal selection algorithm with engineering applications." *Proceedings of Genetic and Evolutionary Computation* Conference: 36-37.

De Castro, L. N. and F. Von Zuben (2000b). "An Evolutionary Immune Network for Data Clustering." SBRN '00 **1**: 84-89.

Farmer, J. D., N. H. Packard, et al. (1986). "The Immune System, Adaptation, and Machine Learning." *Physica* **22**(D): 187-204.

Forrest, S., A. Perelson, et al. (1994). "Self-Nonself Discrimination in a Computer." *Symposium on Research in Security and Privacy*: 202-212.

Hofmeyr, S. and S. Forrest (2000). "Architecture for an Artificial Immune System." *Evolutionary Computation* **7**(1): 45-68.

Jerne, N. K. (1974). "Towards a Network Theory of the Immune System." *Annals of Immunology* **125C**: 373-389.

Kim, J. and P. Bentley (2001). "Towards an Artificial Immune System for Network Intrusion Detection: An Investigation of Clonal Selection with Negative Selection Operator." *Congress on Evolutionary Computation*: 1244-1252.

Nasaroui, O., F. Gonzalez, et al. (2002). "The Fuzzy Artificial Immune System: Motivations, Basic Concepts and Application to Clustering and Web Profiling." *International Joint Conference on Fuzzy Systems*: 711-717.

Timmis, J. and M. Neal (2001). "A resource limited artificial immune system for data analysis." *Knowledge Based Systems* **14**(3-4): 121-130.

Timmis, J., M. Neal, et al. (2000). "An Artificial Immune System for Data Analysis." *BioSystems* **55**(1/3): 143-150.

Watkins, A. (2001). AIRS: A resource limited artificial immune classifier. Department of Computer Science, Mississippi State University. http://nt.library.msstate.edu/etd/show.asp?etd=etd-11052001-102048

Watkins, A. and L. Boggess (2002a). "A new classifier based on resource limited artificial immune systems." *Proceedings of Congress on Evolutionary Computation. Part of the World Congress on Computational Intelligence*.: 1546-1551.

Watkins, A. and L. Boggess (2002b). "A resource limited artificial immune classifier." *Proceedings of Congress on Evolutionary Computation. Part of the World Congress on Computational Intelligence*: 926-931.

Modelling Immune Memory for Prediction and Computation

W. O. Wilson and S. M. Garrett
Computational Biology Group
Dept. Computer Science
University of Wales, Aberystwyth
Wales, UK.
SY23 3PG
{wow93,smg}@aber.ac.uk

Abstract

This paper investigates three models of immune memory: (i) the memory cell model, (ii) the residual antigen model, and (iii) the immune network model. Each model can partially explain how the immune system can remember infections, and respond quickly to re-infections, but we begin to demonstrate that none of them are complete on their own. Our initial appraisal is that all three models should be combined, we make some suggestions about how this might be achieved, and we illustrate our discussion with some tentative results.

1 Introduction

Abstractions of immune memory often focus on the interactions between networks of B cells [1]; alternatively the complexities of immune memory may be somewhat side-stepped by the use of artificial *memory cells* [2]. Immunology agrees that these two mechanisms may exist in nature but, in addition, provides a third possibility: that B cells may be stimulated by residual antigenic material. This mechanism has not been widely examined by the Artificial Immune Systems (AIS) community for its memory properties. This paper considers how immune memory can be modelled in the context of these three mechanisms.

We present some initial results that suggest immune memory can be a simple, emergent property of the interactions between B cells and antigenic material, and their population sizes. However, we also show it is likely that none of the memory mechanisms are sufficient to realistically explain immune memory on their own.

It seems likely that a combination of all three mechanisms is required for robust, realistic immune memory. Furthermore, any combination of these memory mechanisms will exhibit dynamics that are not present in any of the component mechanisms alone. Such a unified model of immune memory would not only be more accurate when predicting the behaviour of the natural immune system, it seems likely to provide new abstractions of immune memory that will have potential in computational systems.

2 Background

At least three models of immune memory exist. It may not be possible, or even necessary, to treat the processes as being separate, but it is useful for our purposes here. They can be described as follows:

The memory cell model: in which memory is mediated by long-lived 'memory cells', which are differentiated from B cells, and which respond almost immediately to the re-presentation of the antigen that led to their creation.

The residual antigen model: in which small amounts of antigen are retained from every infection to continually stimulate the immune system at a low level, keeping it in a state of readiness if the infection (or one similar to it) should it ever return in higher concentrations.

The immune network model: in which lymphocytes not only respond to antigenic regions on foreign bodies, they also respond to antigenic regions on other lymphocytes, such as B cells. This can form a loop of stimulation and repression which continues to stimulate the immune system, even in the absence of the antigenic sequence that originally stimulated the response.

Several questions are raised on consideration of these three models of memory. How are memory cells created, and how do they differ from normal immune cells? How could a small concentration of antigenic material be left behind after an immune response? How can a potentially chaotic network of interactions, between a vast number of immune cells, be controlled and provide useful dynamics? If all three processes do occur in nature, then to what extent to they interact?

3 A Simple Implementation of Immune Memory

Artificial clonal selection algorithms tend to rely on artificial memory cells to 'remember' an infection [2, 3, 4]. They evolve antibodies in response to an antigen, and memory is implemented simply by storing a copy of the best match to the antigen as a memory cell. In contrast, we were interested in the quality and accuracy of the memory response, given various input conditions and sought to investigate how the quality of memory varied given the intensity of an infection.

Initial attempts to model immune memory were based on a unusually detailed clonal selection approach, with a wide variety of parameters controlling aspects of the immune model. The model was implemented using two space concepts: the usual 'shape space' formed by the binding surfaces of antibody and antigen, and a 'body space'. Body space represents the location of antibodies and antigen in an artificial host. The antibody population size, reproduction rate, and mutation factor were varied to explore how they affected the ability of the antibodies to match the antigen. The binding threshold, and recognition region, defined when and where antibodies and antigen could bind. The antigen concentration level, determined the degree of infection. Repeat infections were introduced to explore the AIS's secondary response. We varied various factors with the following results:

The antibody population size had surprisingly little bearing on the quality of the AIS, so long as it exceeded the antigen population by some minimum margin.

The antibody reproduction rate only had a small impact on the AIS, and only if the initial antibody population was a poor fit for the invading antigen, and even then it only delays, not prevents the immune response.

The antibody mutation factor level created two clear regions of activity with very limited middle ground: either the host quickly stabilised the antigen concentration levels and recovered from an infection, or infection increased to the point that the host was overwhelmed (i.e. numbers of antigen became computationally intractable).

The binding threshold had the most significant impact on the AIS. There is a trade-off between higher antigen concentration levels, higher immunity failure rates and longer response periods versus the significant improvement in memory cell quality that is develops over this longer period.

The recognition region, if narrowed, significantly slowed the effectiveness of the AIS, generating much higher antigen concentration levels and longer response times.

The antigen concentration level had a direct influence on the length of the response period, and simultaneously influenced the quality of the memory cells: the longer the response period, the more time was available for the affinity maturation process to fine tune the affinity of the memory cells.

The secondary response time to an infection was considerably shorter than those of the primary response, in the vast majority of cases.

Repeat infections could be repelled, so long as a memory cell was retained with a close affinity to the infecting antigen. Changes in the antibody population between infections were of less significance.

However, these results did not explain how memory cells were formed or maintained, nor how their formation might interact with the other immune memory mechanism mentioned above. This motivated a more biologically realistic approach.

4 Memory Cells as an Emergent Property of the Evolution of Lymphocytes

Each cell in our bodies can reproduce only a predefined number of times, as defined by the length of its *telomeres*, DNA sequences that 'cap' and protect the tips of our chromosomes and which are shorted each time the cell reproduces. What if the degree of telomere shortening were changed in lymphocytes that match antigenic material? And what if that change were proportional to the strength of the match? In that case strongly matching immune cells would tend to survive longer than weakly matching ones.

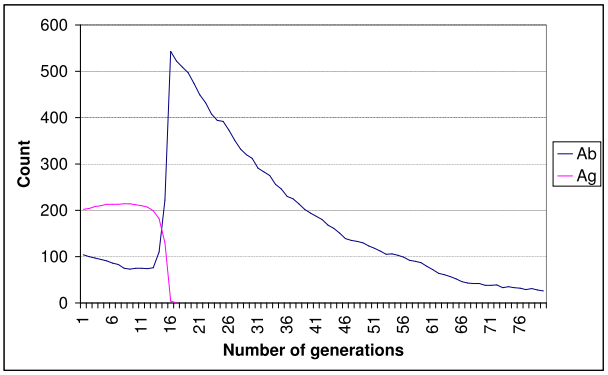

Figure 1: Death rate alone can not maintain a stable population of memory cells. Both prior to the immune response's activation, and after it has ended, the population size is shrinking.

This principle is not new—De Boer has suggested a model based on similar concepts [5]—however, it does not appear to been considered as a memory mechanism in AIS. Dutton, Bradley and Swain agree that the death rate is a vital component required in establishing robust memory. "It stands to reason that activated cells must escape cell death if they are to go on to be memory. Thus, factors that promote the survival of otherwise death-susceptible T cells are candidates for memory factors." [6].

One possibility is as follows. After an infection has been brought under control, and clonal expansion has ended, immune cells die in accordance with their *death rate*, which relates to the length of their telomeres. Crucially, however, cells that are a better match to the infection tend to survive longer, either (i) because their telomeres have not been shortened as much and they retain their length during reproduction, or (ii) because telomerase actually lengthens the telomeres in proportion to the strength of bind with an antigenic region. Here we have assumed (i), but both these possibilities need investigation.

There may also be a difference between naïve cells and memory cells. Grayson, Harrington, Lanier, Wherry and Ahmed state that, "...memory T cells are more resistant to apoptosis than naïve cells ... Re-exposure of memory cells to Ag through viral infection resulted in a more rapid expansion and diminished contraction compared with those of naïve cells." [7].

Another possibility is that the large majority of clonally expanded cells (i.e. effector cells) have their death rates *increased* by the suppression of telomerase for those cells that do not strongly match antigenic sites. This possibility has not yet been investigated.

Figure 1 shows plots of an immune cell population (say, B cells) versus an Ag population, beginning at the point of infection and continuing until several generations after the immune response has ended. The drop in population is entirely caused by immune cell death, and the cells that survive longest tend to be those that were good matches for the infection. Thus, memory cells have emerged, by evolution, from effector cells, without the model making any crisp distinction between the two types of cell.

However, it is clear from Figure 1 that the immune cell population size will eventually fall to zero, because even with a longer lifespan, all cells will eventually die. Although naive cells are being born continually, which can maintain a constant *overall* number of cells, a naive cell can not replace a dying memory cell's memory of an infection—at least, not without first forming an immune response. Therefore, a further mechanism is required to explain how memory cells can be retained, in the long term.

5 Residual Antigen as an Emergent Property of Lymphocyte Function

Several reports suggest that normal lymphocyte function cannot remove all traces of a particular class of antigen. This is a natural result of the immune system being focussed on particular locations in the body. Whilst most antigenic material will be cleared by the immune system, causing a mild immune response, some antigenic material will escape a localised immune response long enough to reproduce. In so doing, the immune system quickly establishes a steady state between immune response and antigenic population size, and the immune system is stimulated by the normal hypermutation response. This continues to preserve memory cells for the given antigen.

Some may ask whether this residual antigen phenomenon may explain immune memory on its own. Perhaps we do not need to concern ourselves with death rates and telomers at all? However, it does not explain why better matching cells tend to survive and worse matching cells tend to die off; nor does it explain how memory cells can naturally emerge as a result of immune cell evolution. Both the apoptosis reduction or maintainance mechanism and the restimulation mechanism are required.

Figure 2 illustrates the effect of re-stimulating the immune response every 4 generations. This was sufficient to keep the population of memory cells from dissipating.

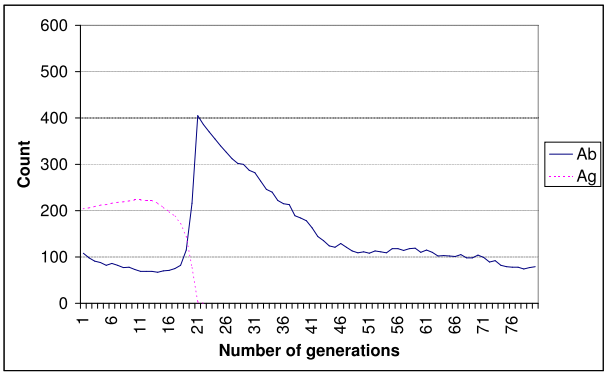

Figure 2: In combination with the residual antigen model, the memory cell model is better able to maintain a stable population of memory cells.

There is another, related possibility. Perhaps memory cells do not need stimulation by antigen; they simply proliferate periodically. Grayson, Harrington, Lanier, Wherry and Ahmed identify the discrepancy between the long term behaviour of memory cells and naïve cells and state that, "... memory cells undergo a slow *homeostatic proliferation*, while naïve cells undergo little or no proliferation." [7] (our emphasis).

Even if re-exposure is not necessary, Antia, Pilyugion and Ahmed conclude " estimates for the half-life of immune memory suggest that persistent antigen or repeated exposure to antigen may not be required for the maintenance of immune memory in short lived vertebrates;however,... repeated exposure may play an additional role in the maintenance of memory of long lived vertebrates" [8].

However, do these two concepts together explain immune memory? To some degree, yes. However, what is not clear is how different immune responses can interact with each other. For this to happen the immune network mechanism is necessary.

6 Immune Network Interactions

The third mechanism is one that is well-studied in AIS: the immune network. The first study of this mechanism was in 1986 by Farmer, Packard and Perelson [9], which defined a set of dynamically changing differential equations for the interactions that might occur between B cells if they were activated by each other, as well as by antigenic binding surfaces. They showed that circuit of stimulation and repression could be formed that would continue even in the absence of antigenic stimulation. However, it does not now seem likely that this mechanism can be the major source of immune memory. So what is the purpose of the interaction between immune cells, assuming that evolution would have otherwise suppressed such a potentially dangerous mechanism?

We suggest that the main purpose of the immune network is to regulate the relationships *between* memories of infections. There is no point storing memories of several, very similar infections when one would cover them all. This is a waste of resources. Since there is a maximum number of memory cells (we have finite volume!) it is evolutionarily sensible to maximise the effect of each memory cell. The immune network would tend to focus its suppressive effects on groups of memory cells that are similar, since this would elicit an immune(-like) response again those cells. This in turn would cause an expansion of other cells, but if those cells are not further stimulated it is likely that their offspring will not form long-lived cells. The result is that the network of memory cells has been kept a lean as possible.

7 Conclusions

7.1 Discussion

Even our initial analysis indicated that the affinity of the resulting memory cells is improved as the number of antigen cells presented to the system increased—since higher antigen concentrations take longer to eliminate, there is more time for affinity maturation to produce high-quality memory cells. However, as the degree of infection increases further, the risk of immune system failure rose and the simple model failed.

Our more detailed study of the three basic hypotheses of immune memory is still in its infancy, but we have already demonstrated that there are many exciting possibilities. We have also shown some tentative results that support our suggestions, and these are supported by biological findings too.

7.2 Future Work

So far, this investigation has been course-grained; future work will refine the following aspects.

The implementation of memory will require extensive evaluation of the various possible combinations of the three memory mechanisms described in Section 2. Their relative contributions will be explored, and the results will be compared to the behaviour of the natural immune system.

Gaussian mutation is currently used to produce new antibodies; a more biological gene library approach should be investigated, to see if it adds realism to the

results.

The models were small scale, involving only a few thousand elements at most. The next development step will increase the number of these elements nearer to computation limits and investigate whether the findings from such an expansion in the model are consistent with our current findings and nature.

The ultimate aim is to produce a model of immune memory that is useful for prediction of immune system dynamics, and to abstract new mechanisms from that model that are computationally able to perform tasks faster, better or uniquely.

References

[1] Timmis, J., Neal, M.J.: A resource limited artificial immune system for data analysis. In: Proceedings of ES2000, Cambridge, UK. (2000) 19–32

[2] de Castro, L.N., Von Zuben, F.J.: Artificial immune systems: Part I—basic theory and applications. Technical Report DCA-RT 01/99, School of Computing and Electrical Engineering, State University of Campinas, Brazil (1999)

[3] Fukuda, T., Mori, K., Tsukiyama, M.: Immune networks using genetic algorithm for adaptive production scheduling. In: 15th IFAC World Congress. Volume 3. (1993) 57–60

[4] White, J.A., Garrett, S.M.: Improved pattern recognition with artificial clonal selection. In: Proceedings of the Second International Conference on Artificial Immune Systems (ICARIS-03). (2003) In preparation; available from the authors.

[5] De Boer, R.J., Noest, A.J.: T cell renewal rates, telomerase, and telomere length shortening. Journal of Immunology (**22**) 5832–5837

[6] Dutton, R.W., Bradley, L.M., Swain, S.L.: T cell memory. Ann. Rev. Immunol **16** (1998) 201–223

[7] Grayson, J., Harrington, L.E., Lanier, J.G., Wherry, E.J., Ahmed, R.: Differential sensitivity to nave and memory cd8+ t cells to apoptosis in vivo. J. Immunol **169** (2002) 3760–3770

[8] Antia, R., Pilyugin, S.S., Ahmed, R.: Models of immune memory: On the role of cross-reactive stimulation, competition, and homeostasis in maintaining immune memory. PNAS **95** (1998) 14926–14931

[9] Farmer, J., Packard, N., Perelson, A.: The immune system, adaptation and machine learning. Physica D **22** (1986) 187–204

PICS: Pittsburgh Immune Classifier System

Alessio Gaspar — IT Department, University of South Florida (USA), http://www.lklnd.usf.edu/, alessio@lklnd.usf.edu

Beat Hirsbrunner — PAI group, CS department, University of Fribourg (Switzerland), http://www.unifr.ch/diuf/pai/, beat.hirsbrunner@unifr.ch

Abstract

A simple computational model of secondary immune response is used to provide a Pittsburgh-style classifier system with the ability to improve its reaction to already encountered situations in a cyclic continuous learning environment. Main results obtained with our core algorithm (YaSais) on Time Dependent Optimization problems are briefly reminded before to introduce the Pittsburgh Immune Classifier System (PICS) which is then experimentally evaluated on both a static and dynamical multiplexer problem. Eventually, the Lazy Optimality Effect, keystone of YaSais' efficiency, is re-examined in PICS. Suggested enhancements are then experimentally evaluated.

1 Introduction

1.1 Motivation, Previous work

Convergence progressively, disables the crossover effects and leaves mutations as the only exploration drive in evolutionary algorithms. Diversity loss is therefore adaptiveness loss. This makes such algorithms poor models of natural adaptive systems when facing ever changing environments.

This led us to rely on Evolutionary Time Dependent Optimization (ETDO) [14, 13, 16] to improve the adaptiveness of a Simple Artificial Immune System [4] in a changing environment. The encouraging results led us to investigate how such a basic algorithm could perform in machine learning problems by being combined with a Learning Classifier System. The underlying concept we want to explore is the notion of *cognitive immunity:* the capability of a machine learning system to learn successive behaviors consecutively suited to a changing environment and recall previously learned behavior when previously encountered situations occur.

1.2 Objective Statements

We want to evaluate a simple, general purpose artificial immune algorithm from Time Dependent Optimization (TDO) tasks to Time Dependent Learning (TDL) problems. Next section introduces the YaSais algorithm [6, 4], sums up previous experimental results and the concept of Lazy Optimality Effect (LOE). Section 3 introduces the Pittsburgh Immune Classifier System (PICS) as a combination of YaSais with a Pittsburgh Classifier System (PCS). Preliminary results on a static and then dynamic multiplexer problem (MUX) are compared to those obtained by YaSais's on TDO. Section 4 further details how specific evolutionary effects present in YaSais introduced unexpected results in PICS. Suggestions as to how to improve PICS are then evaluated. Section 5 concludes by discussing analogies with latent learning classifier systems.

2 YaSais: the core immune algorithm

2.1 Generalities

This section describes YaSais (Yet Another Simple Artificial Immune System), an improved version of Sais algorithm [5], and reviews the most important results (immunization, LOE) needed to ground our later discussion on PICS. To quickly locate YaSais among Evolutionary Algorithms, let's describe it as a Genetic Algorithm which K-Tournament selection has been modified in order to select only some individuals to be cloned and then used to perform exploration (crossover and high mutation rate are applied), and which favors good parents vs. mediocre offspring during recruitment. The main differences are (1) explicit clustering of the population into gatherings, (2) selection of individuals to be cloned while others are kept unchanged (clonal selection) and (3) use of intensive exploration techniques (somatic hypermutation) on clones. To be more accurate, YaSais's key idea is to

divide the population into G equi-sized gatherings of B-Cells 1 . The selection mechanism decides which B-Cell(s) per gathering will be activated and serve as a basis for further exploration. This approach is loosely inspired by Jernes' Idiotypic Networks theory [9, 10] on immune system's memory.
Simply stated, B-Cells 2 can be activated by antigens (when directly useful against one of them) or by other B-Cells (anytime). Therefore, if B-Cell A activates B which activates C which in turn activates A, we have a self reinforcing dynamics. Each B-Cell's activation, and therefore reproduction, is ensured in an endogenic way and memorizing boils down to integrating B-Cells into such idiotypic cycles. Evolutionary Algorithms inspired by this theory bend the evolutionary dynamics so that it is not only convergent but also maintains stable subpopulation with respect to other fitness criteria than optimality in the current environment (e.g. previous optimality in TDO).

2.2 YaSais Algorithm

0. Initialization: create P (0)

- Let P (0) be a population of |P| random B-Cells each λ bits long.
- Arbitrary, P (0) is divided in G groups of B-Cells (Gatherings).
- Generation number t is set to 0.

1. Evaluation

- Compute fitness of each B-Cell in P (t). For the Pattern Tracking, it is the complement of its Hamming distance to the current arbitrary chosen optimum.
- For each Gathering in P (t), mark the best fitted B-Cell. There will be G B-Cells marked in P (t).

2. Clonal Selection: P (t) → Pex

- Create empty population Pex, size |Pex|=G*CF, where CF is the Cloning Factor parameter.
- Fill Pex with G B-Cells by K-Tournaments among the ones marked in P(t).
- Copy each B-Cell in Pex CF times (cloning).
- For each clone in Pex , apply high rate random mutations (hypermutating).

3. Recruitment: P (t) + Pex → P (t + 1)

- For each of the G marked B-Cells in P(t), select a challenger with a K-Tournament (K = 3) in Pex and replace the current B-Cell only if it is less fitted than it.
- Let P (t + 1) = P (t)
- Branch to 1. for a fixed number of iterations.

2.3 YaSais Algorithm step by step

Evaluation Phase
In a Pattern Tracking problem [13], the optimum is arbitrarily chosen as a point of the search space every g generations. The fitness of each B-Cell is therefore measured as its Hamming distance to the current optimum (thus simulating immune-like matching to a given antigen):
$\forall Bi \in P(t)$, $Fitness(Bi) = \lambda - \delta h(Bi, Ot)$
where Bi is the ith B-Cell of P(t), Ot the optimum at generation t, λ the length of its binary code and δh the Hamming distance between two binary strings.
Every $\delta t = 50$ generations (transition period), a new optimum is randomly chosen at a Hamming distance δd from the previous one (transition distance). This evaluation also enables us to mark the n best fitted B-Cells in P(t) (n being an heuristic value).
Pattern Tracking can be seen as the dynamical counterpart of the 0-max problem which has been widely used to understand genetic algorithms. The reasons for choosing this benchmark are twofold. At first, and from a static point of view, it is a simple problem. This helps in keeping experiments focused on the dynamical difficulty and avoid biases induced by other static aspects. Secondly, its parameters can be set to feature a specific dynamical difficulty [3]. This helps evaluating YaSais on well understood and controlled difficulty levels.

Clonal Selection Phase
This phase mimics the core of the immune system's evolutionary dynamics [8]: cloning the B-Cells matching antigens. We pick up the G best B-Cells from P(t) and clone them CF (Clonal Factor parameter) times each in order to obtain the temporary population Pex . Then, we simulate Somatic Hypermutation (Natural Somatic Hypermutation mutates the DNA of B-Cells resulting from clonal selection [8]) by randomly mutating each member of P ex and preserving only the mutants improving fitness.

Recruitment Phase
Eventually, we reintroduce worthy B-Cells from Pex into P(t) in order to build P(t + 1). The B-Cells that have not been involved in the building of Pex remain unchanged so that they can implement an implicit memory of past optima. The marked B-Cells are compared to the winner of a K-Tournament (K = 4) in Pex are only replaced if being less fitted. This approach both guarantees stability of the densities of previous optima which fitness is of no interest anymore, and an elitist dynamics which

forbids the best fitness featured at next generation to be lower than the current one.

2.4 Previous Experimental Results

We briefly sum up previous experimental results obtained with YaSais on a Cyclic Pattern Tracking (PT) problem [13, 16] with a focus on its immunization capability only. In a Cyclic Pattern Tracking (CPT), a list of n successive optima is defined (δt fixed for all). An epoch is a duration of n*δt generations during which all optima are presented. Epochs follow each other and thus enable us to evaluate YaSais' reaction to already encountered transitions. YaSais features a tradeoff between reactivity and robustness [6]. Most evolutionary TDO solutions trade a good robustness for a high fitness level or vice et versa. By comparing YaSais to robust [2, 15] and reactive [17, 1, 7] algorithms, we underlined that YaSais is equivalent in terms of efficiency to methods up to 4 times more computationally expensive which previously proved their superiority to other evolutionary algorithms [4].

YaSais also featured an immunization capability (cf. Figure 1). This experiment was averaged over 50 runs for a length of 1000 generations (4 epochs). YaSais (CF = 4,G = 8, K = 4, |P| = 40, λ = 40, Xc = 0.7, μ = 0.01) was applied to a Cyclic Pattern Tracking problem with 5 optima (δt = 50 and δd = 5 then increased by 5 at each transition).

The upper part of the Figure plots the best fitness per generation. The fitness loss at each transitions is reduced over consecutive epochs which is the sign of an ongoing immunization. In the lower part of the Figure, the densities of the 5 successive optima used in this environment are plotted. This complements the previous information by showing the number of copies of each optimum grow during the period at which is it the current optimum. Moreover, these density curves also show that previous optima are kept in the population (non null density).

2.5 The Lazy Optimality Effect (LOE)

So, YaSais features an immunization capability but a closer look at Figure 1 reveals that, on average, the fitness keeps dropping slightly when previous optima are encountered again. Why aren't all optima durably memorized in the population?

At each generation, |P| - G B-Cells are kept unchanged and G B-Cells are chosen to initiate an intensive search. This mechanism is responsible for loosing previous optima. By taking a closer look to transitions in a single-run experiment (CPT, G=8, |P|=40),) we observed the following pattern: the density of the current optimum decreases suddenly (eg. 6 B-Cells) and keeps doing so (less significantly though) during consecutive transitions.

On the other hand, the density of the next optimum increases from 2 to 8 B-Cells (same example transition). Why are the B-Cells encoding the current optimum more often selected? Quite simply, they are (on average) the closest to the new optimum in the population. Remember that in our CPT problem the n optima are as follow:

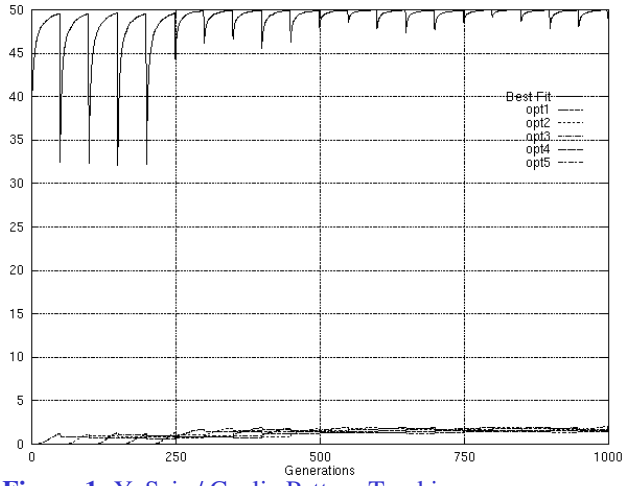

Figure 1: YaSais / Cyclic Pattern Tracking
Top: best fitness, Bottom: 5 optima's densities

B-Cells are divided in λ/n bit-long blocks, the 1st optimum is all '0' except for '1' filling the 1st block, the 2nd optimum has its 2nd block set to '1' and so on. Consequently, the Hamming distance between consecutive optima is constant and equals to 20 bits (n = 5 and λ = 50). We know that YaSais will mark the closest B-Cells to the new optimum. The probability of a random string to match the new optimum is 0.5, and the probability for it to be located at a Hamming distance less or equal to 20 is:

$$P = \sum_{d=0}^{20} 0.5^{\bar{\lambda}}.C_{\lambda}^{d}$$

That is, the probability for a non-previously optimal B-Cell (assumed to be random) to be located closer to the new optimum than any previous optimum is P = 0.1 in our case.

We checked this on a transition in the previous experiment. At generation 1000, YaSais lost 6 B-Cells encoding previous optimum and gained 6 B-Cells encoding new optimum (from 2 to 8). We also counted 17 instanced of current or previous optima. Among the 23 remaining B-Cells, 3 only (P = 0.1) have a chance to be selected instead of previous optima. Knowing that G = 8 are going to be picked up, even if all 3 are retained, 5 out of 6 B-Cells encoding previous optimum should be used for

exploration (6 B-Cells were used). Therefore, if two consecutive optima are close enough, the system forgets about the previous one but still features an overall good performance. Why ? When the distance is short, previous optima are lost. This has limited consequences since finding the new one is simple enough. If the transition gets more difficult, the immunization plays its role. We termed this the Lazy Optimality Effect (LOE), since immunization is only used when nothing simpler works. We replicated previous experiment with only two optima and varied their relative distance to check the influence of this factor. Figure 2 confirms that for a high distance (over 12) the immunization is perfect. Results are also quite good for a very low distance (2) but less good between those two extrema. This is no surprise since a small δd minimizes the fitness loss (cf. supra) but it is important to understand that this is achieved without immunization. Examining the density curves of each optima confirms that with small δd , optima are lost regardless of the misleadingly appealing fitness curve.

2.6 Conclusion: YaSais / TDO

The experimental results on Pattern Tracking revealed that YaSais is an efficient dynamical optimization Tool. A restriction should be kept in mind as we only considered so far non epistasic problems for which we suspect the somatic hypermutation to play a central role in improving the system's reactiveness (cf next section).

YaSais also features an improved robustness to environmental changes; Whenever the transitions are easy (low δd), the natural diversity kept in the population is enough to ensure a good level of fitness to be kept during the transitions. On the other hand, when confronted to difficult transitions, YaSais takes advantage of any relevant information in the population such as previous optima. Therefore, an implicit tradeoff is realized between the use of the random diversity and the "oriented one" induced by the immunization process. The rule seems to be "if it is hard to find, remember it, otherwise, just drop it". Even if not reaching a perfect immunization capability as we initially expected, we must admit that, although it is more "lazy", YaSais uses at best its capabilities.

3 PICS Time Dependent Learning

3.1 PICS algorithm

Classifier Systems (CS) have been investigated in two main flavors. Michigan style CS evolve a population of rules which constitute altogether the

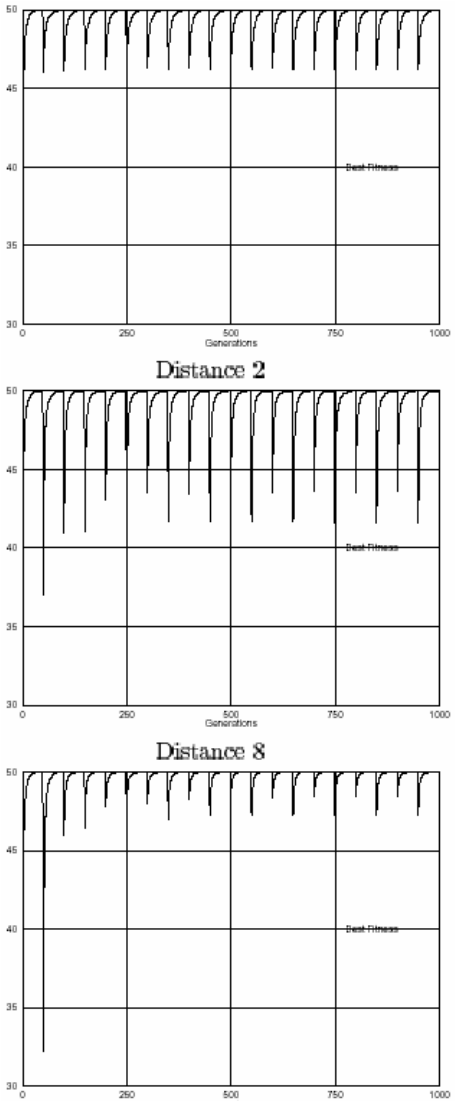

Figure 2: YaSais / Cyclic Pattern Tracking Distance Between Optima vs. Immunization

policy evaluated in a given environment. If a reward is earned, Reinforcement Learning techniques are used to perform the necessary Credit Assignment among the rules that contributed to the successful behavior. On the other hand, the Pittsburgh approach is about evolving a population where each individual encodes the complete ruleset of an independent CS. Fitness is computed by decoding a given individual into a CS and evaluating its interaction with the environment (e.g. average reward over a given time). This approach only relies on evolution to find efficient classifiers and is therefore a natural candidate for the design of a hybrid algorithm embedding the key features of YaSais.

Therefore, we evolved individuals encoding full CS with YaSais instead of a conventional Evolutionary Algorithm. Our objective is to provide a cognitive immunity by preserving previously useful policies in the population. This section details experiments on both static and dynamic multiplexer problems and discusses LOE in a learning context.

3.2 S-7-MUX Experiments

Let us consider a 7 bits instance of the Static Multiplexer Problem (S-7-MUX): we have 6 bits long inputs and 1 bit output. The input is separated in 2 address bits and 4 data bits. For any input, the correct output is the input data bit located at an index given by the decimal value of the 2 input address bits. For instance, input [10 0010] corresponds to output [1]. Consequently, the followings are the minimal and most generic [11] set of rules solving the 7 bits multiplexer problem:

[00 0###] → [0] [10 ##0#] → [0]
[00 1###] → [1] [10 ##1#] → [1]
[01 #0##] → [0] [11 ###0] → [0]
[01 #1##] → [1] [11 ###1] → [1]

Ideally, classifier systems should converge toward this rule set. Basic approaches do not most of the time but recent advances help in ensuring the generality of the solutions [18].
We started off by applying PICS to S-7-MUX with the following experimental conditions:

Experiment: 1300 generations, results averaged over 20 runs B-Cells: $\lambda = 140$ bits encoding 20 rules
Rules: [2 + 4] : [1] (input = 2 bits address + 4 bits data, output = 1 bit)
Population: $|P| = 100$
Evaluations: over 30 input samples (among 64 possible) randomized at each fitness function call
Selection: K = 2 (select) K = 3 (recruit)
PICS specifics: G = 20, CF = 3
Operators: X c = 0.8 (uniform) and $\mu = 0.01$

Figure 3 plots the fitness of the best individual of each generation (upper curve). Knowing that the best reward in this one-step environment is 1000, we can deduce that the approach is performing decently, featuring an asymptotic convergence which is pretty common in evolutionary computation. During single runs, we picked up the best individual at generation 1300 and fed the classifier system it encodes with all 64 possible inputs.
The result of this evaluation of its "coverage" of all perceptions revealed that highly fitted individuals' coverage could be as low as 47%.

To be able to measure this phenomenon reliably and understand it better we decided to measure another statistics during the experiment. The lower curve represents the fitness of the best B-Cell of each generation once computed over 300 samples.
This value is more representative of the true value of each individual and, as can be seen, is lower than the one featured by the quick 30 samples evaluation scheme driving evolution. Let us keep this issue in mind and move on to the other experiments. Section 4 will revisit these observations, suggest and evaluate a solution.

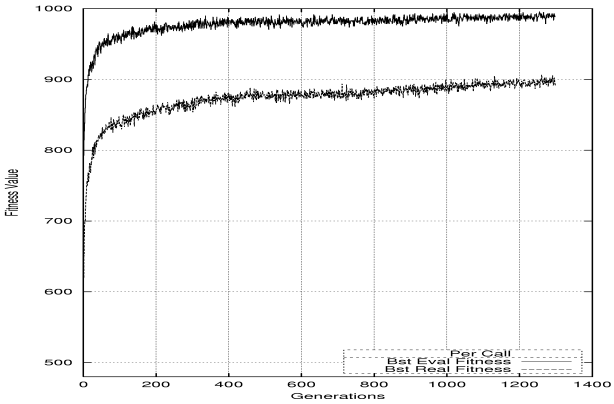

Figure 3: PICS / S-7-MUX
Top: Evaluated Fitness, Bottom: Real Fitness

3.3 D-7-MUX Experiment

This section completes the previous experiment by evaluating PICS immunization capability in a dynamical environment. The dynamical 7 bits multiplexer problem (D-7-MUX) is similar to its static counterpart. We decided to have a transition period $\delta t = 2000$ to allow full convergence. Four different environments are going to be presented during one epoch (8000 generations) and then repeated over and over for 4 epochs (32000 generations). The first environment is S-7-MUX. Then, we generated 3 other environments from it by adding a shift value σ when decoding the address bits. During first period, $\sigma = 0$ then $\sigma = 1$ and so on up to $\sigma = 3$ after which $\sigma = 0$ again as we start a new epoch. Consequently, input address bits 00 will correspond to the 1st input data bit during 1st period, then to the 2nd during 2nd period and so on. Let's see how input [00 0101] is multiplexed over time:

σ= 0 [00 0101] → [0]
σ= 1 [00 0101] → [1]
σ= 2 [00 0101] → [0]
σ= 3 [00 0101] → [1]

Figure 4 also plot the best fitness per generation as evaluated by PICS (upper curve) and accurately

evaluated over 300 samples (lower curve). The vertical dotted lines represent transitions from one epoch to another. Other parameters were kept identical to previous experiment. The following observations can be made:

Immunization:

PICS is indeed able to get immunized to previously encountered optima. Both fitness curves progressively reduce their drop off at transitions to new optima over epochs. PICS' core algorithm therefore turned out to be able to feature an identical immunization ability for both TDO and TDL problems which is the first point we wanted to make sure of in this paper.

Resuming Learning:

Both best fitness curves, but especially the lower one, increase from epoch to epoch. After reaching a certain fitness level while solving the first environment ($\sigma = 0$), PICS deal with 3 other environments. When it is again dealing with the first one, its immunization, besides increasing robustness, also enables it to use the $\delta t = 2000$ generations of the period to improve its fitness level in this environment. It seems to do so from epoch to epoch, "resuming" its learning of each successive optima each time and giving an overall asymptotic trend of improvement.

Fitnesses Differences:

It can also be noticed that the difference between both fitness curves tends to reduce asymptotically over epochs. It can be said that despite the problem underlined in the previous section, PICS manages to overcome it over time as it accumulates information about its environment over epochs instead of converging and discarding any information while re-converging toward another optimum.

4 Corrupted Lazy Optimality Effect

4.1 From LOE to CLOE

Our hypothesis is that the LOE is responsible for the observations in Figure 3.

The upper part of Figure 5 illustrates how B-Cells should be specialized. Let us consider one gathering in P (t). Once the fitness function is computed for all its B-Cells, one is activated (selected to be copied into Pex). This B-Cell will be replaced by a better fitted offspring resulting from the exploration performed in Pex. This can be seen as the gathering getting its best fitted B-Cell furthermore specialized to fit the problem at hand. When environment changes, another individual will be specialized to meet its requirements or a previously activated one re-used thus loosing part of its previous specialization (LOE).

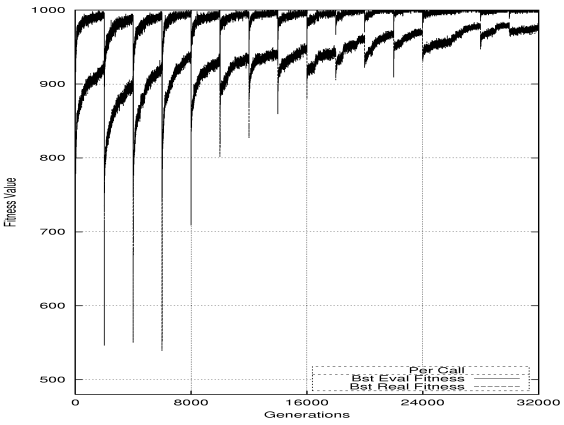

Figure 4: PICS / D-7-MUX
Top: Evaluated Fitness, Bottom: Real Fitness

In practice, the situation is worse since the evaluation sample is randomized at every evaluation thus increasing the bias in comparison to the above example. Next section discusses how to handle this Corrupted Lazy Optimization Effect (CLOE).

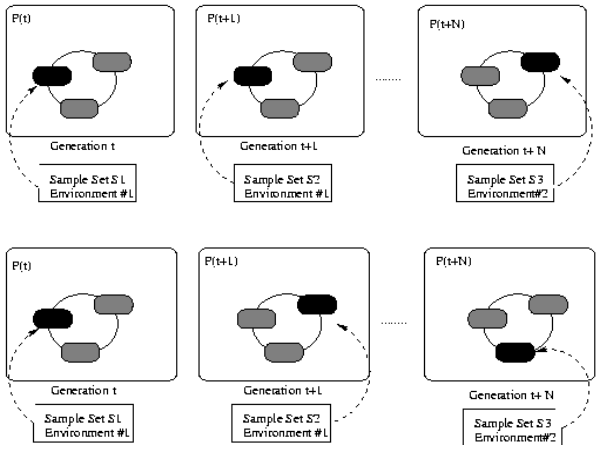

Figure 5: CLOE: specialization of B-Cells
Top: expected activation, Bottom: observed one

The lower part of Figure 5 illustrates what happens in practice when changing the evaluation set every generation. As can be seen, this boils down to changing the fitness function and PICS reacts by specializing another B-Cell (LOE). Conceptually, this is right insofar that, from the evolutionary algorithm standpoint, Time Dependent and Stochastically Evaluated fitness landscapes are the same: fitness values are altered over time. What causes such a change does not make much qualitative difference even though it may influence difficulty of transitions [3]. Nevertheless, we would like PICS to differentiate between changes in the

environment, which call for specialization, and bias due to the stochastic nature of evaluation.

4.2 Getting to know CLOE better

As previously stated, if we take the best B-Cell produced by a run and evaluate it on all possible 64 inputs, its efficiency is way inferior to what its fitness value promised. This can be seen by taking the best B-Cell of each generation and evaluating its fitness over 300 samples instead of 30. This suggests that a whole gathering may be able to react correctly to all possible inputs but a single B-Cell is not. While we expected B-Cells of a gathering to specialize into successively optimal policies, it seems they specialized in solving subsets of all possible input samples. How can we help PICS to specialize only during transitions ? Our working hypothesis is that changing the evaluation set at each generation (as illustrated in Fig. 5) or at each fitness call (as done by PICS) makes a difference. We checked it by changing the stochastic evaluation policy accordingly and decided to randomize the evaluation samples set at each generation and use it for evaluating the whole population.
The top plotting in Figure 6 is similar to Fig. 3. Experimental conditions were identical (S-7-MUX) except concerning evaluation policy. The following observations can be made:

Convergence Time: It has been shortened from 1400 to 400 generations thus providing the algorithm with a fastest handling of static problems.
Fitnesses Differences: The sampled fitness values converge sooner toward the ones obtained with a thorough evaluation. This should lead to a better accuracy in efficiency of evolved policies.

Our second hypothesis is that the more two consecutive evaluation sets differ, the more likely it is for another B-Cell to be activated (cf. LOE).
Therefore, we introduced the overlap parameter: the number of samples kept unchanged from one generation to the next in the evaluation set.
The first plotting in Figure 6 had a null overlap (new evaluation sets at each generation), the second has a maximal value (29 samples are kept unchanged over 30). Results indicate that increasing this parameter degrades efficiency and further separate the real fitness value from the one computed by PICS internally. This clearly invalidates our hypothesis and leads us to conclude that the best evaluation policy is to use the same evaluation set for the whole population and change it completely at each generation to maximize the diversity of samples the system learns from.

5 Conclusion

5.1 Discussion

This paper was first submitted to ICARIS 2002 with the intent to show that a simple AIS could be used along with a classifier system to achieve cognitive immunity. PICS learns continuously policies suited to its current environment, reacts to changes, and keeps previously good policies memorized just in case the environment might be cyclic. When re-occurrences of previous environments occurs, a secondary immune response takes place by triggering a faster (immediate in some cases) recovery of the adequate policy thus implementing the expected so-called cognitive immunity.
The learning problem we investigated in this paper is simple yet highly epistasic, and requires individual to undergo a stochastical evaluation adding noise to the determination of their fitnesses during evolution. Because of these characteristics, it constitutes a good benchmark for YaSais itself, independently of the machine learning aspects.
An interesting analogy can also be drawn with recent advances from the Classifier System community concerning latent learning approaches.
Latent Learning Classifier Systems do not only seek for an optimal policy in a given environment but also build progressively a model of the environment which is improved by every trial no matter how wrong or right it is [12]. PICS also models successively optimal policies which, combined altogether, describe the whole environment dynamics. This information could be used to improve evolved classifier systems or simply understand how the system came to its solution(s).

5.2 Synthesis

This paper presented an hybrid algorithm combining an immune algorithm (YaSais) with a Classifier System. The so-called Pittsburgh Immune Classifier System (PICS) has been evaluated in both a static and dynamic Time Dependent Learning (TDL) environment based on the 7 bits multiplexer problem. Preliminary experimental results revealed that PICS features a secondary immune response in its way to discover and memorize optimal policies for various environments. A particular evolutionary effect has been given more attention, explaining efficiency and suggesting a new improvement which was detailed and evaluated on a static environment.

Acknowledgment: Swiss National Science Foundation grant #20-65301. Originally published in ICARIS 2002 proceedings

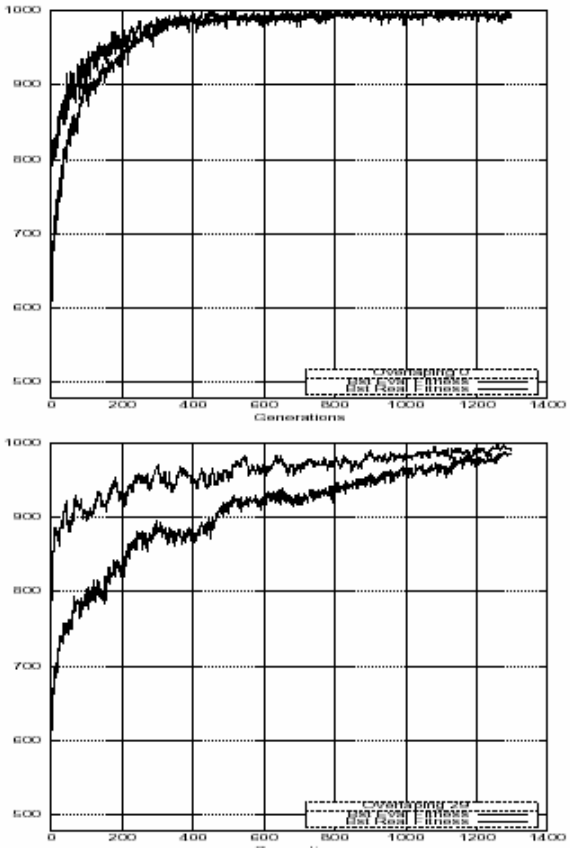

Figure 6: PICS / S-7-MUX
New Evaluation Scheme (overlap parameter)

References

[1] H.C. Cobb and J.J. Grefenstette. Genetic algorithms for tracking changing environments. In Icga-5, 1993.

[2] J.C. Culberson. Genetic invariance: A new paradigm for genetic algorithm. Technical Report 92-02, University of Alabama, 1992.

[3] A. Gaspar. Etude de l'adaptativite de systemes evolutionnaire en environnement a fitness dynamique. In Ph.D. Dissertation, University of Nice Sophia Antipolis, July 2000, 2000.

[**4**] A. Gaspar. Secondary immune response for time dependent optimization. Technical re-
port, PAI group, DIUF, University of Fribourg (Switzerland), July 2002. 23p.

[5] A. Gaspar and P. Collard. From GAs to Artificial immune systems: Improving adaptation in TDO. In CEC-1999: IEEE Congress on Evolutionary Computation. IEEE society press, 1999.

[6] A. Gaspar and P. Collard. Two models of immunization for time dependent optimization. In SMC-2000: IEEE International Conference on Systems, Man and Cybernetics. Special Track on Artificial Immune Systems. IEEE society, 2000.

[7] J.J. Grefenstette. Genetic algorithms for changing environments. In R. Manner and B. Manderick, editors, Parallel Problem Solving from Nature 2, pages 465--501. Elsevier Science Publishers B.V., 1992.

[8] Ron Hightower, Stephanie Forrest, and Alan S. Perelson. The Baldwin effect in the immune system: Learning by somatic hypermutation. In Richard K. Belew and Melanie Mitchell, editors, Adaptive Individuals in Evolving Populations: Models and Algorithms, pages 159--167. Addison Wesley, Reading, MA, 1996.

[9] N.K. Jerne. Towards a network theory of the immune system. Annals of Immunology, 125(C):373--389, 1974.

[10] N.K. Jerne. Idiotypic networks and other preconceived ideas. Immunological Reviews, (79):5--24, 1984.

[11] Tim Kovacs. XCS Classifier System Reliably Evolves Accurate, Complete, and Minimal Representations for Boolean Functions. In Roy, Chawdhry, and Pant, editors, Soft Computing in Engineering Design and Manufacturing, pages 59--68. Springer-Verlag, London, 1997.

[12] W. Stolzmann P. Gerard and O. Sigaud. Yacs: a new learning classifier system using anticipation. Journal of Soft Computing : Special Issue on Learning Classifier Systems.

[13] K. Pettit and E. Swigger. An analysis of genetic based pattern tracking and cognitive based component tracking models of adaptation. In Proceedings of National Conference on AI (AAAI-83), pages 327--332. Morgan Kaufmann, 1983.

[14] R. Salomon and P. Eggenberger. Adaptation on the evolutionary time scale: A working hypothesis and basic experiments. In Evolution Artificielle, pages 297--308, 1998.

[15] R.E. Smith, S. Forrest, and A.S. Perelson. Searching for diverse, cooperative populations with genetic algorithms. Evolutionary Computation, 1(2):127--149, 1993.

[16] F. Vavak and T.C. Fogarty. Comparison of steady state and generational genetic algorithms for use in nonstationary environments. In IEEE International Conference on Evolutionary Computation (ICEC), pages 192--195, 1996.

[17] F. Vavak, T.C. Fogarty, and K. Jukes. A genetic algorithm with variable range of local search for tracking changing environments. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, Parallel Problem Solving from Nature -- PPSN IV, pages 376--385, Berlin, 1996. Springer.

[18] Stewart W. Wilson. Classifier fitness based on accuracy. Evolutionary Computation, 3(2):149--175, 1995.

Immune Systems, Danger Theory and Intrusion Detection

Jamie Twycross
University of Nottingham
jpt@cs.nott.ac.uk

Abstract

This short paper outlines the scope and aims of the work that will be carried out over the next three years in a multidisciplinary EPSRC-funded Adventure Project involving the University of Nottingham, University of the West of England, University College London, Hewlett-Packard Labs, Bristol, and developers of the Firestorm intrusion detection system. The project aims to investigate biological mechanisms proposed by a new paradigm in immunology, Danger Theory, and to apply these mechanisms to build an intrusion detection system which is able to effectively detect misuse of computer networks and systems in real-world environments.

1 Introduction

As society's reliance on computer systems and networks for communication and commerce increases, the need to protect this infrastructure grows ever more important. The field of intrusion detection seeks to build systems which are able to offer such protection, but is currently only able to offer partial solutions due to the complex and dynamic nature of modern computing environments. In recent years computer scientists have turned to biological systems, particularly the human immune system (HIS), in the hope that, if the mechanisms that these systems employ can be understood and transfered into the digital domain, more comprehensive solutions can be found.

The work we intend to carry out over the next three years follows this lead and will we hope deepen our insight into how both the HIS functions and how problems related to coverage and scalability currently encountered when applying these biological mechanisms can be overcome. At an immunological level, we will perform a series of wet experiments which should help clarify how the HIS is able to regulate its responses in the face of a dynamically changing body. By taking these findings and applying them to the intrusion detection problem, we hope to build an intrusion detection system (IDS) which is able to effectively identify intrusions *and* scale to meet the demands of modern computing environments.

This paper begins by giving a broad overview of current approaches and problems in intrusion detection, and then discusses attempts in recent years to build IDS's which take inspiration from the HIS. It then moves on to briefly outline a new paradigm in immunology, Danger Theory, and to discuss how this paradigm could be applied to build IDS's that are able to effectively detect misuse in real-world networks and systems.

2 Intrusion detection systems

IDS's are software systems designed to identify and prevent the misuse of computer networks and systems. There are a number of different ways to classify IDS's, and this section focuses on a distinction based on the manner in which an IDS identifies potential system misuse. Broadly speaking, there exist two approaches in this respect: **misuse detection** and **anomaly detection** [4]. The misuse detection approach examines network and system activity for known misuses, usually through some form of pattern-matching algorithm. For example, a misuse-based IDS's might examine all the network packets it sees for a sequence of bytes known

to cause a particular server to crash. In contrast, an anomaly detection approach bases its decisions on a profile of normal network or system behaviour, often constructed using statistical or machine learning techniques. For example, an anomaly-based IDS might build up a profile of the commands a user usually executes on their system and then monitor the user's commands for sequences that deviate from this norm, identifying such deviations as system misuse.

Each of these approaches offers its own strengths and weaknesses. Misuse-based systems generally have very low false positive rates but are unable to identify novel or obfuscated attacks, leading to high false negative rates. Anomaly-based systems, on the other hand, are able to detect novel attacks but currently produce a large number of false positives. This stems for the inability of current anomaly-based techniques to cope adequately with the fact that in the real world normal, legitimate computer network and system usage changes over time, meaning that any profile of normal behaviour also needs to be dynamic. The reduction of this false positive rate in anomaly-based systems, while at the same time maintaining a low false negative rate, is one of the three key problems which our research over the next three years will address.

3 Artificial immune systems

The HIS protects the body against damage from an extremely large number of harmful bacteria and viruses, termed **pathogens**, and does this largely without prior knowledge of the structure of these pathogens. This property, along with the distributed, self-organised and lightweight nature of the mechanisms by which it achieves this protection, has in recent years made it the focus of increased interest within the computer science and intrusion detection communities. Seen from such a perspective, the HIS can be viewed as a form of anomaly detector with very low false positive *and* false negative rates.

Much work has been done recently [1] which attempts to understand and extract the key mechanisms through which the HIS is able to achieve its detection and protection capabilities. A number of **artificial immune systems** (AIS's) have been built for a wide range of applications including document classification, fraud detection, and network and host intrusion detection. These AIS's have met with some success and in many cases have rivalled or bettered existing statistical and machine learning techniques. However, in dynamic environments such as complex computer networks, AIS's have failed to scale to match the large quantity of data these environments produce. This is partly due our continued lack of understanding at a biological level of how the HIS actually works, and also partly due to naive algorithmic implementations of processes known to be occuring in the HIS. Moreover, we believe there has been an over-reliance on the SNS Theory, discussed in the next section, and its mechanisms, such as negative selection, which, although important, are computationally expensive and unsuited to data rich scenarios. As part of our research, we will carefully consider, model and balance a range of immune processes and build an immune-based IDS which, as well as having a low false negative and false positive rate, will be able to scale effectively. This is the second key problem our research agenda will address over the next three years.

4 Danger Theory

The question then is, how do we intend to reduce the false positive rate of the anomaly-based IDS we build? We are hoping the answer lies in a relatively new paradigm in immunology - **Danger Theory** [2]. The prevalent view in immunology, the **SNS Theory** [3], is that the HIS is geared to discriminating self from nonself. It is this ability to distinguish self from nonself that determines if it will react in a destructive or tolerant manner to the cells and proteins, called **antigens**, it encounters. To achieve this discrimination, the cells that compose the HIS go through a number of developmental processes and then interact in a complex network once they have fully matured. The final decision as to how to react to an antigen rests with these cells, and the proteins and cells of the body that the HIS protects are essentially considered passive bystanders in this process.

Danger Theory, however, rejects the notion that body is a passive bystander and, instead of placing the decision-making on the cells of the HIS, places it on the entities the HIS protects. This is not to

say that Danger Theory denies the importance of the processes and interactions the SNS Theory has been successful in uncovering - it does not. It simply states that the final say in how the HIS will react rests with the things being protected and *not* with the system doing the protection. Cells are able to guide the response of the HIS through the production of **danger signals** - specific chemicals released by cells when they undergo stress or unprogrammed death, termed **necrosis**, as opposed to **apoptosis**, cell death which occurs as part of the normal functioning of our bodies. This paradigm shift offers potential solutions to problems which have dogged immunology for years, such as how the HIS is able to cope with a dynamic self which changes throughout the lifetime of an individual.

By exploring and drawing inspiration from the underlying mechanisms of Danger Theory we hope to be able to build an immune-based IDS which is able to effectively protect and adapt to dynamically changing environments while maintaining a low false positive rate. However, little is currently know about the exact nature of danger signals and their interaction with the components of the HIS. Therefore, the third key task on our research agenda will be the identification, through a series of wet experiments, of the key signals produced by apoptotic and necrotic cell death, and the interaction of these signals with components of the HIS in the regulation of immune responses.

5 Conclusion

Danger Theory seems promising from an intrusion detection perspective as it appears to explain how a self-organised, distributed and lightweight system - the HIS - is able to effectively protect a dynamic, complex system - the human body - from damage by previously unknown pathogens. If over the next three years we are able to gain sufficient insight into how danger signals regulate immune responses and to identify corresponding signals *in silico*, and if we are able to apply these insights to build an IDS which is able to effectively scale to cope with real-world scenarios, we feel we will have made a significant contribution to the fields of both immunology and intrusion detection.

Acknowledgements

Thanks to Uwe Aickelin for helpful comments. This project is supported by EPSRC (GR/S47809/01), Hewlett-Packard Labs, Bristol, and Gianni Tedesco.

References

[1] Jung Won Kim. *Integrating Immune Algorithms for Intrusion Detection.* PhD thesis, University of London, July 2002.

[2] Polly Matzinger. An innate sense of danger. *Seminars in Immunology*, 10:399–415, 1998.

[3] Ruslan Medzhitov and Charles A. Janeway. How does the immune system distinguish self from nonself? *Seminars in Immunology*, 12:185–188, 2000.

[4] NIST. Intrusion detection systems. NIST Computer Science Special Reports SP-800-31, November 2001.

Category: Outlines of proposals designed to attract collaboration/constructive comments

An Antigen Presenting Cell Modeling for Danger Model of Artificial Immune System

Anjum Iqbal[1] and Mohd Aizaini Maarof[2]

Group on Artificial ImmuNe Systems and Security (GAINS)
Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia
Emails: [1]anjum@siswa.utm.my, [2]maarofma@fsksm.utm.my

Abstract

The Danger Theory (DT) viewpoint outlines a model of immunity based on the idea that the immune system is more concerned with entities that do damage than with those that are foreign [1,2,3]. U. Aickelin and S. Cayzer [4], look at this theory from the perspective of Artificial Immune System (AIS) practitioners. According to them DT is not about the way AIS represent data. Instead, it provides ideas about which data the AISs should represent and deal with. They should focus on dangerous, i.e. interesting data. U. Aicklin et al [5] intend to use the correlation of signals based on the DT. They believe the success of their system to be independent of the eventual acceptance or rejection of the DT by immunologist as the proposed AIS would achieve this by identifying key types of apoptotic and necrotic alerts and understanding the balance between these two types of alerts. In addition, the proposed AIS is extended by employing the Antigen Presenting Cell (APC) activation mechanism explained by the DT. This mechanism has the advantage of detecting rapidly spreading viruses or scanning intrusions at an early stage [5].

As APC plays primary roll in intercepting danger signals and conveying them to environment [1, 4], therefore it is important to look into the details of the mechanisms, exhibited by an APC, in DT viewpoint. The abstractions of these mechanisms might be used to design APCs for DT based AISs. The danger signal might not only be generated for necrosis, perhaps some proactive alert signals the presence of protein intending to produce cell damages or stresses. There might be an alert for the presence of a self but dangerous protein prepared under the instructions of disease susceptible genetic peptide [6].

Taking the first guide line from U. Aickelin and S. Cayzer [4], we intend to initiate our work, for DT based APC modeling, with the analysis of normal (self) data, which bears potential for producing danger. We hope that the detailed analysis of danger susceptible self data should provide us with guide lines for tracing our way for modeling APC mechanisms. The literature of computational biology currently lacks the material regarding DT based modeling. We expect the success of our research in cooperation with the research results and guidance from world renowned AIS research teams.

Keywords:
Artificial immune systems, danger theory, antigen presenting cell, modeling, data analysis

Reference:

1. P. Matzinger (2002), "The Danger Model: A Renewed Sense of Self," Science Magazine, 296: 301-305, 2002.

2. P. Matzinger, "The Real Function of The Immune System." http://glamdring.ucsd.edu/others/aai/polly.html

3. P. Matzinger (2001), "Introduction to the Series," Scand. J. Immunol. 54, 2-3, 2001

4. U. Aickelin and S. Cayzer (2002). "The Danger Theory and Its Application to Artificial Immune Systems." In Proceedings of The International Conference on Artificial Immune Systems (ICARIS), pp. 141-148

5. U. Aickelin, et al (2003). "Danger Theory: The Link between AIS and IDS?" In Proceedings of The International Conference on Artificial Immune Systems (ICARIS)

6. Stacey B. Gabriel et al (2002), "The Structure of Haplotype Blocks in the Human Genome," Science Magazine, Vol. 296, 21 JUNE 2002

An Artificial Immune System for Misbehavior Detection in Mobile Ad Hoc Networks with both Innate, Adaptive Subsystems and with Danger Signal

(work in progress)

Slaviša Sarafijanović and Jean-Yves Le Boudec
{slavisa.sarafijanovic, jean-yves.leboudec}@epfl.ch
EPFL/IC/ISC/LCA,
CH-1015 Lausanne, Switzerland

***Index Terms*— Mobile, ad-hoc, misbehavior, detection, artificial, immune, clonal selection, learning, adaptive, cognitive.**

I. Extended Abstract

A. Problem Statement and Related Work: Detecting Misbehaving Nodes in DSR

The successful operation of a mobile ad hoc network depends on cooperation of the nodes in providing services to each other. Nodes act both as terminals and information relays, and participate in a common routing protocol, such as Dynamic Source Routing (DSR) [13]. The network is vulnerable due to faulty or malicious nodes. Misbehavior detection systems aim at removing this vulnerability [1], [2], [3], [4], [6], [7].

Our approach for misbehavior detection in DSR is to use an Artificial Immune System (AIS) [14], [15]. The system is inspired by the natural immune system of vertebrates [10]. The main task of the natural immune system (the IS) is to protect the human body against microorganism invaders and some malfunctioning own cells, while being tolerant to normal own cells, self cells. To accomplish this task, the IS has developed some detection and reaction mechanisms and procedures, which may be useful for solving analogous problems in building an AIS.

The work presented here is a continuation of our previous work [6], [7]. In the previous work we proposed a solution for mapping some basic parts of the IS to our AIS: representation, matching, and negative and clonal selection. We implemented and validated the solution in the Glomosim simulator [11]. The system had a separate preliminary phase for collecting self-behavior examples. This phase had to be run in a protected environment, when there is no misbehavior of the nodes. It is very hard to provide such conditions in a real network.

In this work we give three main improvements for our AIS. First, we propose a solution that doesn't require a preliminary learning phase in the protected environment (the environment without misbehavior). The solution uses analogy with the IS danger signal [8], [9]. Second, we add the innate part of the AIS, which provides fast detection of misbehavior patterns that are known in advance and for which specific detection mechanisms are designed. For the innate part we adopt solution given in [3]. Third, we introduce information exchange between the nodes, which is analogous to the use of cytokines in the IS; for the information exchange we use the robust reputation scheme proposed in [2]. In our previous work, there was one immune system per network node; with this third improvement, there is one global immune system, distributed across all nodes.

B. Learning Changing Self in an Unprotected Environment. Use of Danger Signal.

The main difficulties for providing self-tolerance in our case are caused by the fact that the system to be protected (mobile ad hoc network running DSR) changes over time. This is because of mobility, changes in nodes' traffic and software updates. The AIS need to learn to differentiate between **new** normal behavior and misbehavior. Our solution for learning changing self, that works well if started in possibly unprotected environment (that may contain misbehaving nodes) is given on Figure 1.

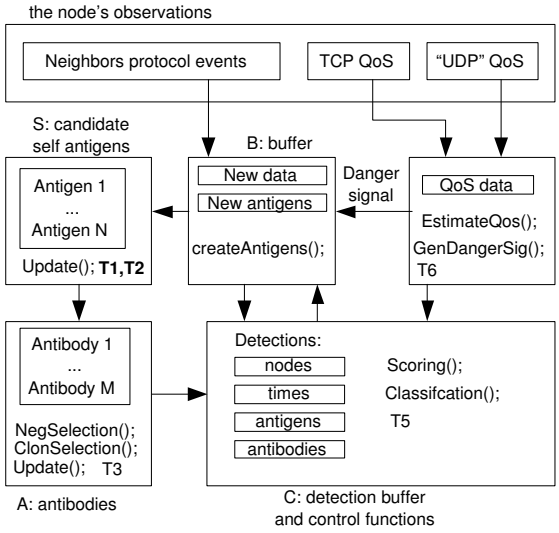

Fig. 1. Learning self antigens in an unprotected environment. The scheme works even if misbehavior is present during both initial and normal operation phase.

The main idea is to use the quality of service (QoS) obtained by a node when is communicating over some neighbors, and correlate it with matching results on the antigens that are describing observed behavior for that neighbors in the near past. QoS measures that we use are throughput of TCP connections and response time of applications which use UDP,

as compared to their estimated normal values. An antigen is created from the data collected during an interval Δt. For more details on representation, matching, and classification see [6], [7].

Here is how the scheme shown in Figure 1 works. Initially, all collected self antigens become 'candidate self antigens', until S becomes full. Then the initial set of antibodies is created. Subsequently collected antigens are buffered in B. They will be checked for matching, and detection results will be temporally stored in C, for both antibodies and antigens. The corresponding node and the time of the detection are also temporarily stored in C. The storing time is determined by T_5 (T_i are system constants). T_5 controls how much time on average are detection results collected for some node, before it is classified as misbehaving.

If there is no matching between the current set of antibodies and an antigen that is collected by the node for one of its neighbors, if no bad QoS is experienced by the node over that neighbor, and if no bad QoS is reported in a sufficient amount for that neighbor by others, in the near past, the antigen will be used for the updating S. The minimal update interval is determined by the constant ΔT; it controls the maximum speed of change of the protected system that may be followed by the AIS, when QoS is good.

The antigens that belong to a node, for which there is enough evidence that it misbehaves, will not be used for updating S. The evidence is calculated from own detections and experienced QoS, and the detections and QoS reported by neighbors. By this distributed filtering, we achieve that S is updated with self antigens. Updates by nonself antigens happen quite rarely, because we use additional latency T_1 (in addition to T_5) in updating the set of self antigens; this time constant is larger than the time needed by the system to detect the node which generated it, unless we have persistently good QoS in the near past (T_4). A nonself antigen that passes this barriers and deletes antibodies reactive to it will also be detected and eliminated from S, but only by the correlating it to bad QoS, and after a longer time, and then again the antibodies will be created that contain knowledge of this antigen and speeds up the detection.

There are two types of antibodies: normal, with T_3 half life time, and memory, that has an infinite life time. Normal antibodies die if they are not useful in detection for some time. Our AIS deletes self-reactive memory antibodies, unlike in the IS case. If a memory antibody consistently matches antigens collected during good QoS in the neighborhood, it will be deleted. In this way, we solve the problem of chronical auto-immunity that is usually caused by mimicry between self and experienced nonself antigens. Such a solution is not used by the IS, because of antigen presenting cells APC and lymphocyte trafficking constraints [10].

C. Using both the innate and the adaptive part

The innate part of the natiral IS has fast detection and reaction against some pathogens with known nonself patterns on their surface. For some misbehavior types, the innate part may detect that an attack is maybe going on, but it has no appropriate detection and reaction to resolve the problem. In both cases it signals to the adaptive part, mobilizing more resources of the adaptive part. This signaling is important because some of attacks are not solved by the innate part, but by the adaptive part or by a cooperative effort.

The part of our solution described in Section I-B is the adaptive part of our AIS. We add the innate part by coding mechanisms that directly detect events which refer to misbehavior or possibility of misbehavior. Such events are non-forwarding route request or data packets, and some unallowed changes in protocol fields in relayed packets. Our innate system influences the adaptive system in that the adaptive systems reacts more quickly when there is evidence that the innate part has detected anomalies. This is analogous to battlefield cytokines in the natural IS.

D. Distributed AIS. Cooperation of nodes. Information exchange

Detection, classification and QoS information are exchanged between the nodes adopting the reputation system proposed in [2], [3]. This provides faster gathering the evidence needed for safe classification of nodes as misbehaving, and reaction against them. It also changes our analogy to the natural IS in that the body to be protected is now the entire network instead of nodes in isolation.

E. Model Validation

We implement and validate our model in the ns-2 simulator [12]. We show improvements over previous work in time to response, ability to detect new attacks, and false positive ratios.

REFERENCES

[1] Sergio Marti, T.J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of MOBICOM 2000*, pages 255–265, 2000.

[2] S. Buchegger and J.-Y. Le Boudec. A Robust Reputation System for Mobile ad hoc Networks. Technical Report IC/2003/50, EPFL-DI-ICA, Lausanne, Switzerland, July 2003.

[3] S. Buchegger and J.-Y. Le Boudec. Performance Analysis of the CONFIDANT protocol: Cooperation of nodes - Fairness In Distributed Ad-Hoc Networks. In *Proceedings of IEEE/ACM Symposium on Mobile Ad-Hoc Networking and Computing (MobiHOC)*, Lausanne, CH, June 2002. IEEE.

[4] S. Buchegger and J.-Y. Le Boudec. The Effect of Rumor Spreading in Reputation Systems for Mobile Ad-hoc Networks. In *Proceedings of WiOpt '03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks"*, Sophia-Antipolis, France, March 2003.

[5] S. Buchegger, Cedric Tissieres, J. Y. Le Boudec "A Test-Bed for Misbehavior Detection in Mobile Ad-hoc Networks - How Much Can Watchdogs Really Do?" Technical report No. IC/2003/72, November 2003.

[6] J. Y. Le Boudec and S. Sarafijanovic. An Artificial Immune System Approach to Misbehavior Detection in Mobile Ad-Hoc Networks. Proceedings of Bio-ADIT 2004, Lausanne, Switzerland, January 2004, pp..

[7] S. Sarafijanovic and J. Y. Le Boudec. An Artificial Immune System Approach with Secondary Response for Misbehavior Detection in Mobile Ad-Hoc Networks. TechReport IC/2003/65, EPFL-DI-ICA, Lausanne, Switzerland, November 2003.

[8] P. Matzinger. Tolerance, Danger and the Extended Family. *Annual Review of Immunology*, 12:991-1045, 1994.

[9] P. Matzinger. The Danger Model in it's Historical Contex. *Scandinavian Journal of Immunology*, 54:4-9, 2001.

[10] L.M. Sompayrac. How the Immune System Works, 2nd Edition. Blackwell Publishing, 2003.

[11] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. Glomosim: A library for parallel simulation of large scale wireless networks. *Proceedings of the 12th workshop on Parallel and Distributed Simulations-PDAS'98*, May 26-29, in Banff, Alberta, Canada, 1998.

[12] The network simulator ns-2, http://www.isi.edu/nsnam/ns/.

[13] D.B. Johnson and D.A. Maltz. The dynamic source routing protocol for mobile ad hoc networks. *Internet draft, Mobile Ad Hoc Network (MANET) Working Group*, IETF, February 2003.

[14] De Castro, L. N. and Von Zuben, F. J. (1999), "Artificial Immune Systems: Part I Basic Theory and Applications", Technical Report RT DCA 01/99.

[15] Leandro N. de Castro and Jonathan Timmis, "Artificial Immune Systems: A New Computational Intelligence Approach" , Springer Verlag, Berlin, 2002.