Original paper

Soft Computing 7 (2003) 526–544 © Springer-Verlag 2003
DOI 10.1007/s00500-002-0237-z

Artificial immune systems as a novel soft computing paradigm

L. N. de Castro, J. I. Timmis

Abstract Artificial immune systems (AIS) can be defined as computational systems inspired by theoretical immunology, observed immune functions, principles and mechanisms in order to solve problems. Their development and application domains follow those of soft computing paradigms such as artificial neural networks (ANN), evolutionary algorithms (EA) and fuzzy systems (FS). Despite some isolated efforts, the field of AIS still lacks an adequate framework for design, interpretation and application. This paper proposes one such framework, discusses the suitability of AIS as a novel soft computing paradigm and reviews those works from the literature that integrate AIS with other approaches, focusing ANN, EA and FS. Similarities and differences between AIS and each of the other approaches are outlined. New trends on how to create hybrids of these paradigms and what could be the benefits of this hybridization are also presented.

Keywords Framework, Artificial immune systems, Hybrid intelligent systems, Survey of hybrids

1 Introduction

The increase in the understanding of several computational intelligence approaches, such as artificial neural networks, fuzzy systems, evolutionary algorithms and probabilistic reasoning (PR), led to the proposal of the soft computing (SC) paradigm (Bonissone, 1997; Novák, 1998). SC is usually viewed as a fusion of these approaches, which in turn provides foundations for the conception, design and development of computational intelligence systems. By combining or hybridizing such paradigms, it has been possible to create a number of successful and sophisticated solutions to complex real-world problems.

Each of the established soft computing paradigms contributes their own set of particular characteristics. The main contribution of FS is as an approach for dealing with imprecision, approximate reasoning, information granulation and computing with words. ANN major uses are concerned with system identification, learning, generalization, local search and adaptation. The paradigm of EA is one of systematized search, tuning and optimization. Finally the main contribution of PR is for decision analysis and management of uncertainty.

Zadeh (1997) argued that SC is not just a mixture of approaches, rather it is a partnership, in which each strategy contributes a distinct methodology for addressing problems in its domain. Indeed, this is one of the aspects argued in this paper. AIS can contribute with the already established methodologies in order to mutually improve their performances and application domains.

There has been a growing interest in the use of metaphors extracted from the immune system for the development of the so-called artificial immune systems. AIS have been applied to a wide variety of domain areas, such as pattern recognition and classification (Hunt and Cooke, 1996; Carter, 2000; de Castro and Timmis, 2002b), optimization (Fukuda et al., 1998; de Castro and Von Zuben, 2000a), data analysis (Timmis and Neal, 2001; de Castro and Von Zuben, 2000b), computer security (Kephart, 1994; Kim and Bentley, 1999; Hofmeyr and Forrest, 2000) and robotics (Ishiguro et al., 1997; Jun et al., 1999).

To date, there are two edited volumes on AIS (Ishida et al., 1998; Dasgupta, 1999) that present collections of papers detailing works on theoretical immunology and attempts at constructing AIS. Survey works for AIS do exist (Dasgupta, 1999; de Castro and Von Zuben, 2000c), however, there has been no review presented focusing the hybrids of AIS with other computational intelligence paradigms. There is no work in the literature that tries to highlight and draw together important processes involved in the development of AIS. Additionally, no effort has been made in the direction of presenting a general framework to the design of artificial immune systems.

This paper attempts to address these and other deficiencies in the literature: (1) to review the fundamentals of immunology for the development of AIS, (2) to present a general framework in which to design artificial immune systems, (3) to review works combining AIS with other SC paradigms, (4) to highlight some similarities and differences between AIS and other SC approaches, and (5) to

Published online: 14 July 2003

L. N. de Castro* (✉), J. I. Timmis**
* School of Computer and Electrical Engineering (FEEC),
State University of Campinas, Brazil. C.P., CEP: 13083-970
e-mail: lnunes@dca.fee.unicamp.br
** Computing Laboratory, University of Kent,
Canterbury, Kent, CT2 7NF, UK
e-mail: J.Timmis@kent.ac.uk.

Leandro N. de Castro would like to thank the Computing Laboratory and CNPq (Profix n. 540396/01-0) for the financial support and Prof. Dr. Fernando J. Von Zuben for his indispensable comments on the development of a framework for the AIS. Jon Timmis would like to thank the Computing Laboratory, UKC for their continued support in this new area of research.

discuss new ways of integrating AIS with other SC approaches.

The paper is organized as follows. Section 2 introduces the basic immunology background necessary to develop and understand AIS. Section 3 introduces the field of AIS and proposes a general framework to design AIS. Section 4 puts places AIS in context with other SC paradigms by outlining some similarities and differences of these different strategies. In Sect. 5 a survey of hybrids of AIS with ANN, EA and FS is presented. Section 6 summarizes the outcomes of integrating AIS with SC techniques and suggests new avenues to the creation of hybrid models of AIS with ANN, EA and FS. The paper is concluded in Sect. 7 with a discussion of the main contributions of the AIS to the SC community.

2
Fundamentals of the immune system

All living beings are endowed with an immune system whose complexity varies according to their characteristics. For example, some plants have spines to provide protection from predators to that attack them. Animals containing bones (vertebrates) developed a highly effective and comple x immune system. It is composed of a vast array of cells, molecules and organs that work together to maintain life. The focus here will be on the immune system of vertebrates, more specifically of humans. This is due to its interesting features, from a biological and computational perspective, the great knowledge available about its' functioning and its broad applicability in the design of AIS.

The immune system performs several functions. Together with other bodily systems it maintains a stable state of our vital functions, named homeostasis. Its most remarkable roles however, are the protection of the organism against the attack of disease causing agents, called *pathogens*, and the elimination of malfunctioning cells.

Microorganisms like viruses, bacteria, fungi and parasites are classified as pathogens, for they can cause diseases after invading our bodies. The primary problem the immune system is faced with, is thus the *recognition* of these pathogens. The pathogens themselves cannot be directly recognized by the components of the immune system. Some small portions of the pathogens, named *antigens*, are the molecules that are going to be recognized by the immune system. After recognizing (identifying) a disease causing agent, the immune system is responsible for eliminating it, so as to avoid or block the disease.

There are a few other tasks however, that the immune system has to perform so that it can correctly identify and eliminate pathogens. One such task is the recognition of our the body's own tissues, which are broadly named *self*. Like pathogens, the cells and molecules of our the body's organisms also present antigens, in this case *self antigens*, that can be recognized by the immune system. In order to distinguish self antigens from those presented by pathogens, the latter are named *nonself antigens*. The process of distinguishing between self and nonself antigens (i.e. what belongs and what does not belong to the body) is termed *self/nonself discrimination* and will be discussed in a dedicated subsection.

The discussion of the immune system will begin by highlighting the main types of immune cells and molecules, and how they are generated. This follows with a description of a mechanism used to explain how the immune system copes with nonself antigens, coupled with the self/nonself discrimination problem. Finally, a network theory of the immune system is presented. These topics were chosen, as they inspired the development of a number of works composing the proposed framework used to develop artificial immune systems.

2.1
Physiology of the immune system

There are two organs responsible for the generation and development of immune cells: the *bone marrow* and the *thymus*. The bone marrow is the site where all blood cells are generated and where some of them develop. The thymus is the organ to which a class of immune cells named T-cells migrates and maturates.

There are also several types of immune cells, but the focus will only be concentrated on the *lymphocytes*. Lymphocytes are white blood cells specialized mainly in the *recognition* of pathogens. There are two main types of lymphocytes: B-cells and T-cells, both originated in the bone marrow. Those lymphocytes that develop within the bone marrow are named B-cells, and those that migrate to and develop within the thymus are named T-cells. Both cell types present *receptor molecules* on their surfaces responsible for recognizing the *antigenic patterns* displayed by pathogens or some of their fragments. Figure 1 illustrates a B-cell and a T-cell with their surface receptor molecules detached. Note that both cell types have a number of identical cell receptors on their surfaces. The T-cell receptor is called TCR and the B-cell receptor is called BCR or *antibody* (Ab).

2.1.1
The bone marrow

The bone marrow is a soft tissue located in the cavity of the most elongated bones. Specifically, it is the site where all blood cells are generated and some of them differentiate into B-cells.

During the differentiation of a blood cell into a B-cell, it produces and displays an antibody molecule on its surface, as illustrated in Fig. 1a. An antibody molecule has two main functions: (1) to bind with (recognize) an antigen, and (2) to perform an effector function.

Figure 2 illustrates that the antibody molecule is composed of two main regions in a Y-shaped form. One major

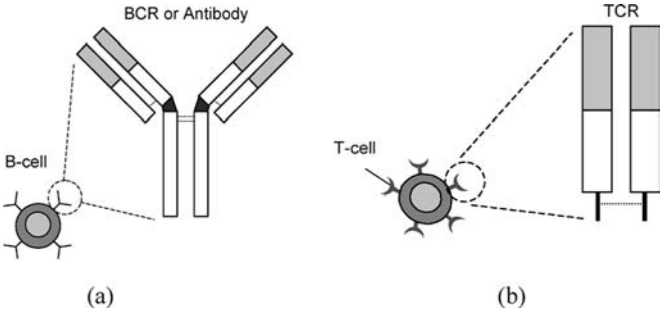

Fig. 1. Lymphocytes and their surface receptor molecules. **a** B-cell receptor (BCR) or antibody. **b** T-cell receptor (TCR)

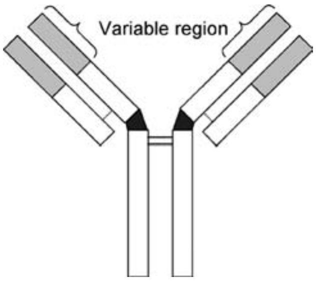

Fig. 2. Antibody molecule detaching its variable region, i.e. the portion of the molecule responsible for recognizing antigens

region named *variable region*, is highly variable and is responsible for the recognition of antigens. The other region named *constant region*, can assume a few different types and is responsible for attaching the antibody to the cell surface and to perform effector functions. Antibodies are originally attached to the B-cell surface but can be released to the blood stream during an immune response.

Antibody molecules are generated in the bone marrow through processes of DNA rearrangement. Genes contained in several gene libraries are concatenated to form the antibody molecules (Tonegawa, 1983). A simplified view of how an antibody molecule is generated from a set of genes collected from gene libraries is illustrated in Fig. 3.

2.2
Pattern recognition in the immune system

Pattern recognition in the immune system occurs basically at the molecular level. The surface receptors of B-cells and T-cells present a certain "shape" that has to be matched by the shape of an antigen, as illustrated in Fig. 4. There are other features that are involved in the recognition of an antigen by a cell receptor, however these are outside the scope of this paper.

Both B-cells and T-cells present surface receptors for antigens. The distinguishing features between them are the basic structures of the receptors (antibodies and TCRs) and the types of antigens each one is able to recognize. While antibodies can recognize and bind with antigens free in solution, TCRs can only recognize and bind with antigens presented by molecules of our own body, known as *major histocompatibility complex* (MHC). TCRs thus, recognize molecules known as peptide/MHC complexes, as illustrated in Fig. 4b. Peptides are simply processed (broken down) portions of the antigen.

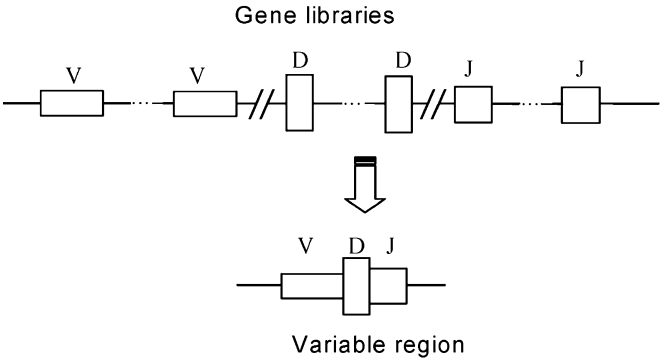

Fig. 3. Simplified view of how an antibody molecule is generated in the bone marrow. One gene segment from each library is selected and concatenated together with others in order to form portions of an antibody molecule

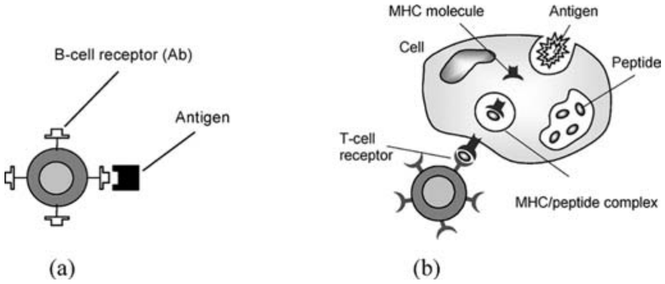

Fig. 4. Pattern recognition in the immune system. **a** Recognition of an antigen by a B-cell receptor. Antibodies can recognize antigens free in solution. **b** Recognition of an MHC/peptide complex by a TCR (T-cell receptor). In order for a TCR to recognize an antigen, it has to be broken down into fragments (peptides) and presented to the TCR by an MHC molecule

It is important to note that the recognition in the immune system is based on shape complementarity. Antigens and cell receptors have to have complementary shapes so that they can bind together. It is the binding together of the receptor with the antigens that trigger an *immune response*, i.e. the reaction of the immune system against the pathogen that displays the antigen recognized.

2.3
Clonal selection

The paper has already discussed how the immune cells are generated and how they recognize antigens. The question now remains: what happens then after recognition?

After successful recognition the *adaptive immune response* is elicited. One important immune mechanism of defense is to reproduce those cells capable of recognizing and binding with antigens. The cellular reproduction in the immune system is based on cloning (mitosis), i.e. the creation of offspring cells that are copies of their parent cells subject to mutations. This proliferation will result in the production of a *clone* of cells of the same type. Due to the mutations, the cells within a clone are all similar but present slight differences and are capable of recognizing the antigen that triggered the immune response. A selective mechanism guarantees that those offspring cells (in the clone) that better recognize the antigen, which elicited the response, are selected to have long life spans; these cells are named *memory cells*. This is the strategy by which evolution shaped our immune systems so that they became capable of dealing with antigens it has encountered in the past. This is also the principle used for vaccination purposes. The whole process of antigen recognition, cell proliferation and differentiation into memory cells is named *clonal selection* (Burnet, 1959; Ada and Nossal, 1987) and is summarized in Fig. 5.

2.3.1
Affinity maturation

The term *affinity maturation* is given to the combined processes of mutation that affect the portions of the receptor that bind with the antigen and the selection that

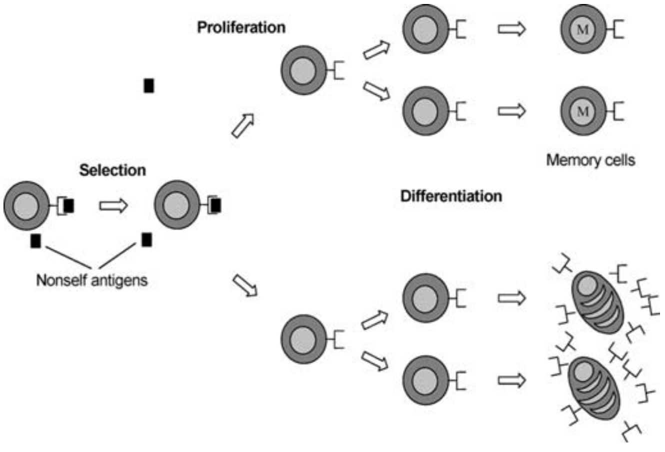

Fig. 5. *Clonal selection*. Those immune cells whose receptors recognize and bind with a nonself antigen are selected to proliferate (subjected to mutation) and differentiate into long living (memory) cells

guarantees the survival of the variant offspring that better match the antigen. *Affinity* refers to the degree of binding of the cell receptor with the antigen. The higher the affinity the stronger the binding and thus the better the immune recognition and response. This immune response is said to be adaptive because it allows, through mutation followed by selection, the cell receptors to adapt themselves to antigens. This guarantees that successive encounters with a certain type of antigen leads to ever more powerful responses against this antigen.

As the lymphocytes of the immune system are somatic cells, i.e. cells not involved in the reproduction of the organism, the mutation that occurs during affinity maturation is named *somatic mutation*. In addition, the rates of mutation that occur during cloning (reproduction) have high rates, suggesting the terminology *somatic hypermutation*. Somatic hypermutation is believed to be inversely proportional to the cell affinity: the higher the affinity a cell receptor has with an antigen, the lower the mutation rate and vice-versa (Berek and Ziegner, 1993). This is another strategy the immune system found to preserve the high affinity variants (offspring cells) at the same time it offers a higher probability of generating major variants of the receptor selected.

Inversely to the somatic mutation, the proliferation rate of a cell is directly proportional to its affinity with the antigen. When a nonself antigen invades an organism, some a number of immune cells recognize this antigen with different degrees of affinity. These cells then undergo clonal selection and affinity maturation. The number of offspring a cell has is proportional to its affinity with the antigen: the higher the affinity, the higher the number of offspring generated and vice-versa.

Clonal selection affect both B-cells and T-cells, but affinity maturation has only been observed in B-cells.

2.4 Self/nonself discrimination

If the immune system is capable of recognizing any antigenic pattern (shape) that is the complement of the immune cell receptors, how does the immune system behave when it is confronted with a self antigen? The answer to this question is rather complex, controversial and involves different mechanisms for B-cells and T-cells. Due to the focus of this paper, the discussion will be restricted to the *thymic negative selection of T-cells*. This is the process responsible for eliminating all T-cells whose receptors recognize and bind with self antigens presented in the thymus (Nossal, 1994; Mannie, 1999).

The thymus is an organ located in the upper region of the chest and to which some white blood cells (naïve T-cells) migrate after being produced by the bone marrow. These *immature* or *naïve T-cells* will then suffer a process of *negative selection* within the thymus. A blood thymic barrier avoids nonself antigens to be present within the thymus. Thus all antigens present in the thymus are self antigens. As a consequence, the naïve T-cells that recognize self antigens within the thymus are purged from the population of T-cells. The naïve T-cells that do not recognize any self antigen become immunocompetent T-cells, i.e. T-cells capable of performing an immune response. These are then released to the blood stream and patrol the body in the search for nonself antigens (presented by MHC molecules). This simplified process of negative selection in the thymus is illustrated in Fig. 6.

2.5 Immune network theory

Clonal selection is the theory used to explain how the immune system responds to nonself antigens. Negative selection in contrast, is one of the strategies used by the immune system to eliminate self-reactive cells, i.e. cells that recognize self antigens. There is still another crucial question to be answered: how does the cells of the immune system interact with other cells of the immune system?

Intrigued by this problem, Jerne (1974) proposed a network theory for the immune system. This theory is difficult to be proven experimentally; a reason why it has been refuted by some immunologists (e.g. Langman and Cohn, 1986). Despite the controversies about the validity of this *immune network theory*, it is very interesting and

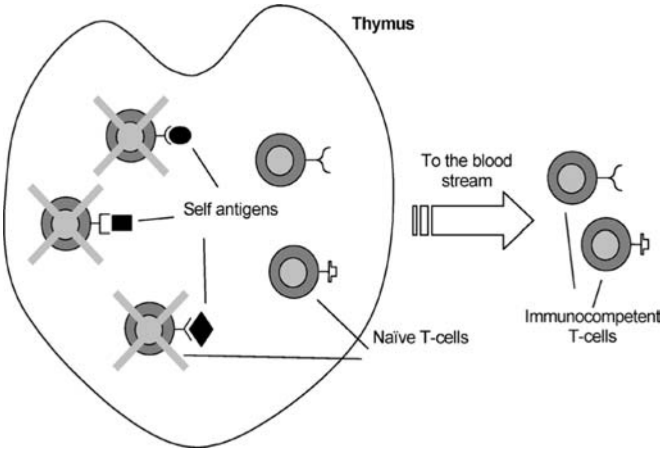

Fig. 6. Simplified view of the thymic negative selection. The naive T-cells that recognize self antigens are purged from the repertoire. In contrast, those that do not recognize any self antigen become immunocompetent cells and are released to the lymph and blood stream

has been widely used in the development of artificial immune systems.

Simply stated, the immune network theory suggests that antibody molecules have portions of their receptors that can be recognized by other antibody molecules. This way, antibodies cannot only recognize nonself antigens but also self antigens, i.e. those presented by other antibodies. As an outcome of this mutual recognition of antibody molecules, a network of communication arises within the immune system; this is called the *immune network*. According to this new perspective of the immune interactions, a nonself antigen is no longer necessary to promote a dynamic behavior in the immune system. The interactions of the immune cells are going to result in a network with a natural eigen-behavior whose state will be disturbed by nonself antigens. An antibody Ab_1 that recognizes a nonself antigen Ag is also capable of recognizing another antibody Ab_2. As both Ag and Ab_2 are recognized by the same antibody Ab_1, Ab_2 is said to be the *internal image* of the antigen Ag. Figure 7 summarizes the principles of the immune network theory.

3 Artificial immune systems

The establishment of the field of *artificial immune systems* (AIS) has been difficult for a number of reasons. First, the number of people active in the research area is still small, but has been increasing in the past few years. Secondly, most researchers found it difficult to identify the difference between an AIS and work undertaken in theoretical immunology. Thirdly, the application domains of artificial immune systems are very wide range. Finally, only very recently the first textbook proposing a general framework to design AIS has been published.

There were a limited number of attempts to define the field of artificial immune systems. The present work adopts the concept in which artificial immune systems are defined as computational systems inspired by theoretical immunology and observed immune functions, principles and models, applied to solve problems (de Castro and Timmis, 2002a). This definition covers some of the aspects mentioned above by drawing a fine line between AIS and theoretical immunology: the applicability. While works on theoretical immunology are usually aimed at modeling and providing a better understanding of the immune functioning and laboratory experiments, works on AIS are applied to solve problems in computing, engineering and other research areas as well. This is more akin to a soft computing paradigm.

The proposal of a framework to design artificial immune systems fills some of the gaps necessary to answer the remaining questions. Initially we have to give our own viewpoint of what we mean by framework. Assume the case of other soft computing approaches inspired by biology, such as ANN and EAs.

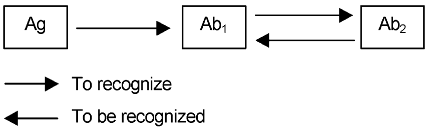

Fig. 7. Summary of the immune network theory. Ab_1 recognizes Ag and Ab_2, thus Ab_2 is said to be the internal image of Ag. Ab_1 and Ab_2 recognize each other mutually

A set of artificial neurons can be arranged together so as to form an artificial neural network. In order to acquire some knowledge, these neural networks suffer an adaptive process, named learning or training, that alters (some of) their free parameters. Thus, in a simplified form, a framework to design an ANN is composed of a set of artificial neurons, a pattern of interconnection for these neurons and a learning algorithm.

In evolutionary algorithms, there is a set of "artificial chromosomes" representing a population of individuals that will iteratively suffer processes of reproduction, genetic variation and selection. As a result of this adaptive process, a population of evolved artificial individuals arises. A framework, in this case, would correspond to the genetic representation of the individuals of the population, plus the procedures for evaluation, reproduction, genetic variation and selection.

Therefore, in our viewpoint, a framework to design a computationally inspired algorithm requires, at least, the following basic elements:

- *A representation for the components of the system*;
- *A set of mechanisms to evaluate the interaction of individuals with the environment and each other*. The environment is usually simulated by a set of input stimuli, one or more fitness function(s), or other mean(s);
- *Procedures of adaptation* that govern the dynamics of the system, i.e. how its behavior varies over time.

This is the basis of the proposed framework to design artificial immune systems as well: a representation to create abstract models of immune organs, cells and molecules; a set of functions, termed affinity functions, to quantify the interactions of these "artificial elements", and a set of general purpose algorithms to govern the dynamics of the AIS. Figure 8 summarizes the elements involved in the framework to engineer an AIS: this can be thought of as a layered approach of the design procedure.

3.1 A representation scheme

In Sect. 2, B-cells and T-cells were described as some of the most important cells in the immune system. It was noted

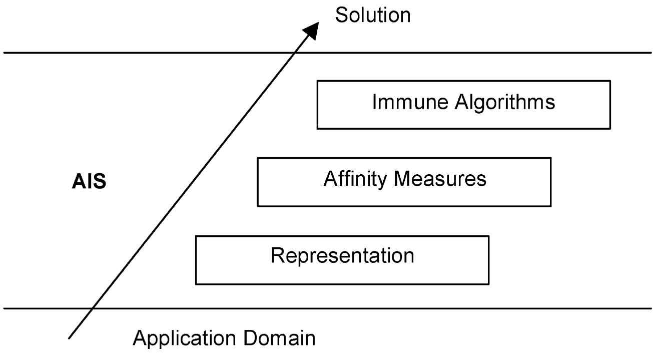

Fig. 8. The framework to engineer AIS and its layered structure

that they present surface receptor molecules whose shapes are complementary to the shapes of antigens, allowing them to recognize the disease-causing agents and then perform an effector function. The immune cells and molecules are therefore the elements that have to be modeled and used to create AIS.

Perelson and Oster (1979) first proposed the concept of *shape-space* (S). Bearing in mind that the recognition of antigens is performed by the cell receptors, shape-spaces allow a quantitative description of the interactions of receptor molecules and antigens. As in the biological immune system, in a shape-space S, the *degree of binding* (*degree of match* or *affinity*) between an antigenic receptor (Ab or TCR) and an *antigen* (Ag), is measured via *regions of complementarity*. This is illustrated in Fig. 9.

The set of features that describe the relevant properties of a molecule from a recognition perspective is termed its *generalized shape*. The generalized shape of an antibody is described by a set of L parameters. Thus, a point in an L-dimensional shape-space, S^L, specifies the generalized shape of an antibody binding region with regard to its antigen binding properties.

A population (repertoire) of N individuals (cell receptors) corresponds to a shape-space with a finite volume V containing N points. As the antigen-antibody interactions are measured via regions of complementarity, the antigenic determinants are also characterized by generalized shapes whose complements lie within the same volume V. This binding between antigen and antibody can be thought of, in simple terms, as a lock and a key. A perfectly complementary key to the lock will open it. However, it is possible to find a key that will insert into a lock, but not open it. In this situation there is a lower affinity between the two; the *bind* is not complete. If the antigens and antibodies are not quite complementary, then the two molecules may still bind, but with lower affinity.

It is assumed that each antibody specifically interacts with all antigens whose complements are within a small surrounding region. This region is characterized by a parameter ε, called *affinity threshold*. The volume V_ε resulting from the definition of the affinity threshold is called *recognition region*. It is possible that an antigen may present some different forms, that is, be a slight variation of the same antigen. As each antibody can recognize all antigens whose complements lie within V_ε, it holds that a finite number of antibodies can recognize an almost infinite number of points (Ag) within the volume V_ε. This is related to *cross-reactivity* (Smith, 1997; Hodgkin, 1998), where similar patterns occupying neighboring regions of the shape-space can be recognized by the same antibody shape, as far as an adequate ε is provided. Figure 10 illustrates the shape-space S, detaching antibodies, antigens and the affinity threshold.

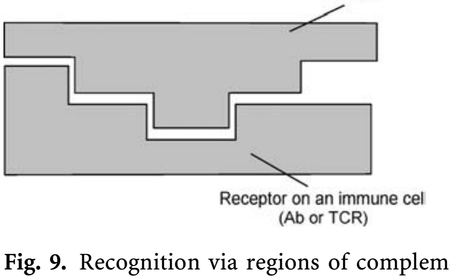

Fig. 9. Recognition via regions of complementarity

3.1.1
Shape-spaces and affinity measures

The type of shape-space (representation) used to model an antibody and an antigen will partially determine a measure to calculate their *affinity*. Mathematically, the generalized shape of a molecule (m), either an antibody (Ab) or an antigen (Ag), can be represented as an attribute string (set of coordinates) $m = \langle m_1, m_2, \ldots, m_L \rangle$, $m \in S^L \subseteq \mathscr{R}^L$, or other more elaborate structures such as a neural network or a Petri net. The discussion here will focus only on attribute strings. For more complex structures, please refer to de Castro and Timmis (2002a).

As the Ag–Ab affinity is related to their distance, it can be estimated via any distance measure between two strings or vectors, such as the Euclidean, the Manhattan, or the Hamming distance. Hence, if the coordinates of an antibody are given by $\mathrm{Ab} = \langle Ab_1, Ab_2, \ldots, Ab_L \rangle$ and those of an antigen are given by $\mathrm{Ag} = \langle Ag_1, Ag_2, \ldots, Ag_L \rangle$, then the distance D between them can be defined as:

$$D = \sqrt{\sum_{i=1}^{L} (\mathrm{Ab}_i - \mathrm{Ag}_i)^2} \ , \tag{1}$$

$$D = \sum_{i=1}^{L} \left| \mathrm{Ab}_i - \mathrm{Ag}_i \right| \ , \tag{2}$$

$$D = \sum_{i=1}^{L} \delta_i, \quad \text{where } \delta i = \begin{cases} 1 & \text{if } \mathrm{Ab}_i \neq \mathrm{Ag}_i \\ 0 & \text{otherwise} \end{cases} \ , \tag{3}$$

where Eq. (1) is the Euclidean distance, Eq. (2) the Manhattan distance and Eq. (3) the Hamming distance.

Given a representation for the molecules, the shape-space formalism defines a space S with a finite volume V in which all molecules are represented. If we assume a given antigen to be recognized, it is possible to introduce the concept of an *affinity landscape* as a representation of the space of all possible affinities of the antigen-binding sites (antibodies or TCRs) in relation to this antigen, as illustrated in Fig. 11.

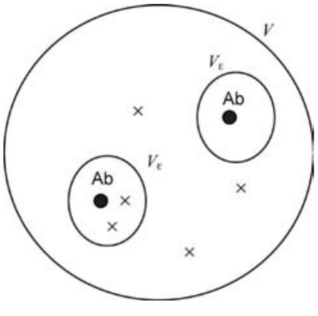

Fig. 10. In shape-space S, there is a volume V in which the shape of an antibody (•) and that of the complement of an antigen (×) are located. An antibody can recognize any antigen whose complement is situated in a volume V_ε around it (after Perelson, 1989)

Many current implementations of AIS assume a binary shape-space, i.e. a shape-space in which binary strings (bitstrings) represent the molecules. In this case, the affinity between an antibody bitstring and an antigen bitstring can be determined by using several different measures. The most widely used are: (1) Hamming distance (Eq. 3), (2) *r*-contiguous bit rule, and (3) multiple *r*-contiguous bit rule.

The Hamming distance can be computed by applying the exclusive-or operator (XOR) to the binary strings (Forrest and Perelson, 1992; Hajela, and Lee, 1996; Hightower, et al., 1996). The *r*-contiguous bit rule measures the number of *r*-contiguous complementary symbols (Forrest et al., 1994; Dasgupta, and Forrest, 1996) between two strings. In the multiple *r*-contiguous bit rule (Hunt, and Cooke, 1996), extensive complementary regions are supposed to be interesting for the detection of similar characteristics in symmetric regions of the molecules. It is defined according to Eq. (4).

$$D = D_{\mathrm{H}} + \sum_i 2^{l_i} \ , \tag{4}$$

where D_{H} is the total Hamming distance given by Eq. (3), and l_i is the length of each complementary region i with at least two consecutive complementary bits. The computation of these three affinity measures is illustrated in Fig. 12.

3.1.2
Generating the initial repertoires

The immune cells and molecules are generated in the bone marrow. The genes used to encode the receptor molecules are stored in separate and distinct libraries. The encoding of these molecules occurs through the concatenation of different gene segments that are randomly selected from each of the gene libraries. Bone marrow models are used to create the attribute strings that represent the immune receptors. Note that up to now, no distinction was made between a cell and its receptor; both are represented by attribute strings. This is mainly due to the fact that each immune cell presents several receptors of same shape on its surface. While discussing immune network models, more complex cell models will be reviewed.

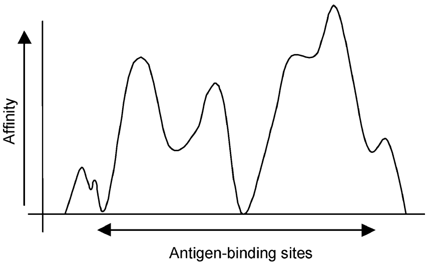

Fig. 11. Pictorial affinity landscape for an Euclidean shape-space of dimension 2

The simplest bone marrow model is the one that generates attribute strings of length L in S^L using a (pseudo-) random number generator. In the case of real-valued shape-spaces, one has simply to determine the interval in which m is going to be defined, e.g. $m \in [0, 1]^L$. In the case of Hamming shape-spaces, the string that represents m must be composed of elements belonging to a pre-defined alphabet, e.g. $m \in \{0, 1\}$ for binary strings (bitstrings).

The more sophisticated, and biologically appealing, models to construct repertoires of immune cells, demand the use of gene libraries from which the molecules will be rearranged or evolved. Hightower, et al. (1995), Perelson, et al. (1996) and Oprea (1999) employed a genetic algorithm (GA) to study the effects of evolution in the genetic encoding of the antibody molecules. One characteristic of this encoding is that not all genes existent in the *genotype* (total collection of genes) are expressed in the *phenotype* (expressed antibody molecules). In these models, libraries of gene segments contain the genes that will be recombined to the generation of the antibody molecules, as depicted in Fig. 13.

Another important feature of this model is that with a relatively small number of genes in the libraries, a large number of different receptor molecules (attribute strings) can be generated. Assume that the AIS contains l libraries, each of which with c components, then c^l antibody molecules can be generated.

3.2
Algorithms and processes

The first steps for the design of an AIS have now been described. It was suggested that an attribute string is, in most cases, a suitable representation for a cell receptor and an antigen. Attention was then given on, the evaluation of their interactions, and how to generate an initial population of strings using a bone marrow model. The type of attributes will partially define a function to evaluate the interactions (quantify recognition) of the cell receptors with the environment (antigens) and each other (other cell receptors). The next step in the framework corresponds to the application of some (usually) iterative procedure of adaptation that will govern how the AIS will behave over time.

The algorithms resulting from the modeling of the processes described in Sect. 2 have been widely used, and sometimes slightly modified and then applied to problem

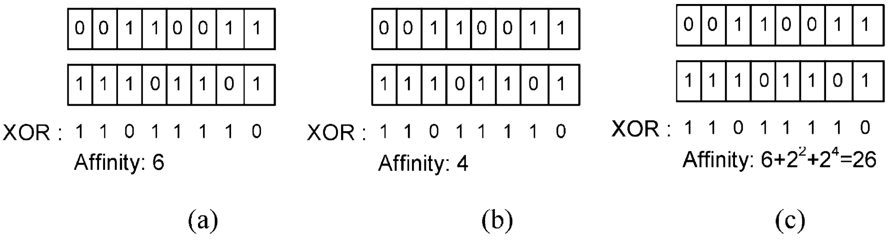

Fig. 12. Affinity measures for the binary Hamming shape-space. **a** Total number of complementary bits (Eq. 3). **b** *r*-contiguous bit rule. **c** Multiple *r*-contiguous bit rule (Eq. 4)

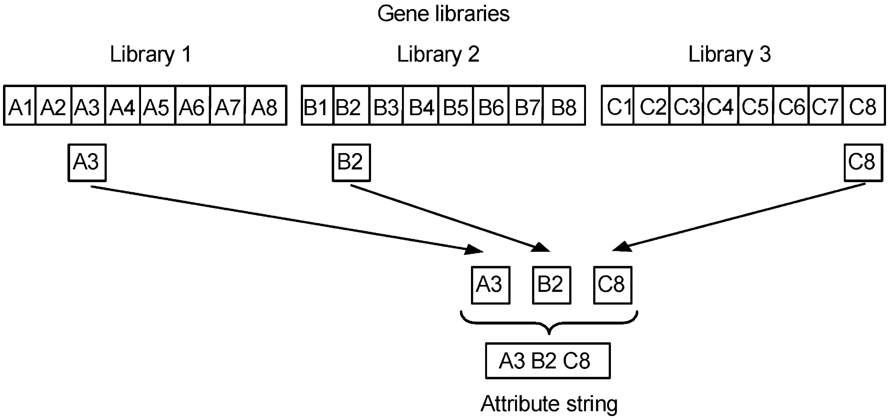

Fig. 13. Process of synthesizing an antibody molecule from gene libraries. One component of each library is concatenated with others to create an attribute string that represents an immune receptor

solving. Thus, they currently constitute some of the basic iterative procedures of adaptation employed in AIS. To make the exposition clearer, we propose the following taxonomy for AIS algorithms: population-based and network-based (Fig. 14). We classify clonal and negative selection algorithms as population-based, and network models are divided into continuous and discrete.

3.2.1
Clonal selection

In de Castro and Von Zuben (2000a) the authors focused on the clonal selection principle and affinity maturation process of the adaptive immune response in order to develop an algorithm suitable to perform tasks such as machine learning, pattern recognition, and optimization. Their algorithm was evaluated in a simple binary character recognition problem, multimodal optimization tasks and a combinatorial optimization problem; more specifically the travelling salesman problem (TSP). The main immune aspects taken into account to develop the algorithm, named CLONALG, were: selection and cloning of the most stimulated cells proportionally to their antigenic affinity; death of non-stimulated cells; affinity maturation and selection of cells proportionally to their antigenic affinity; and generation and maintenance of diversity. The algorithm CLONALG works as follows:

1. Generate a set of N candidate solutions (antibody repertoire) in a shape-space to be defined by the problem under study;
2. Select n_1 highest affinity cells in relation to the antigen set to be recognized or to the function being optimized;
3. Clone (generate identical copies of) these n selected cells. The number of copies is proportional to their affinities: the higher the affinity, the larger the clone size (number of offspring);
4. Mutate with high rates (hypermutation) these n selected cells with a rate inversely proportional to their affinities: the higher the affinity, the smaller the mutation rate (see further discussion);
5. Re-select n_2 highest affinity mutated clones to compose the new repertoire;
6. Replace some low affinity cells by new ones;
7. Repeat steps 2 to 6 until a given stopping criterion is met.

The authors characterized CLONALG as an evolutionary-like algorithm with the main features of population-based search guided by the mechanisms of reproduction, genetic variation and selection. It is important to note however, that though CLONALG is a type of evolutionary algorithm, it was developed using inspiration from the immune system. In contrast, the standard evolutionary algorithms were devised inspired by the neo-Darwinian theory of evolution. Thus, in the former case (CLONALG) the evolutionary theory is used to explain how the algorithm behaves, and in the latter case (EAs) the evolutionary theory was used to create the algorithm.

There are however, some important differences between CLONALG and a GA for example. CLONALG performs not only affinity proportionate selection, but also affinity proportional mutation, and there is no crossover. Similarity does exist however, in the fact that both algorithms encode the individuals of the population. When compared with the evolution strategies (Rechenberg, 1994), for example, again, differences exist between the algorithms. Evolution strategies work with real-valued encoding, while CLONALG works with binary representation, and the affinity proportional mutation in CLONALG is not controlled by Gaussian distributions. Therefore, no matter which type of evolutionary algorithm is compared with CLONALG, there are always enough differences between them, in terms of inspiration and computation that justify the proposal of CLONALG as an evolutionary algorithm inspired by immunology.

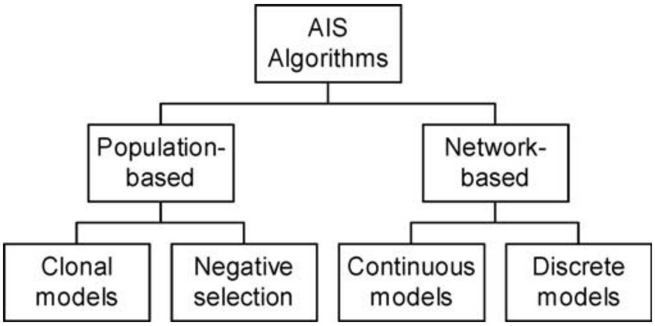

Fig. 14. A taxonomy for AIS algorithms

3.2.2 Affinity maturation

In order to promote and maintain diversity of antibodies, the immune system employs a mechanism known as *somatic hypermutation*. This mechanism also allows the immune system to increase the affinity (recognition capability) of the antibodies in relation to the selective antigens; a process named *affinity maturation*.

As the shape-space formalism allows for the representation of any cell receptor and antigens through attribute strings, it is possible to use several algorithms to insert variations in the encoding of these components. These algorithms can be the same as the mutation operators employed in evolutionary algorithms (e.g. single- and multi-point mutation for bitstrings, and inductive mutation for real-valued vectors), providing the appropriate shape-space is respected.

One important aspect of the somatic hypermutation is that each candidate solution (attribute string) will have an independent mutation rate proportional to its affinity with the nonself antigen. Thus, candidates in higher peaks of the affinity landscape will be subject to smaller mutation rates while candidates located far from optima solutions will suffer larger mutation rates. The idea behind this approach is that candidates close to a local optimum must be fine-tuned, while candidates far from an optimum can perform large steps towards an optimum or another region of the affinity landscape.

However, one problem with this approach is that usually, nothing is known *a priori* about the optima solutions of a function (or problem). In this case, one can evaluate the relative affinity at each time step of each candidate by scaling (normalizing) their affinities. The inverse of an exponential function can be used to establish a relationship between the hypermutation rate α and the normalized affinity D^*, as described in Eq. (5) and depicted in Fig. 15.

$$\alpha(D^*) = \exp(-\rho D^*) \ , \qquad (5)$$

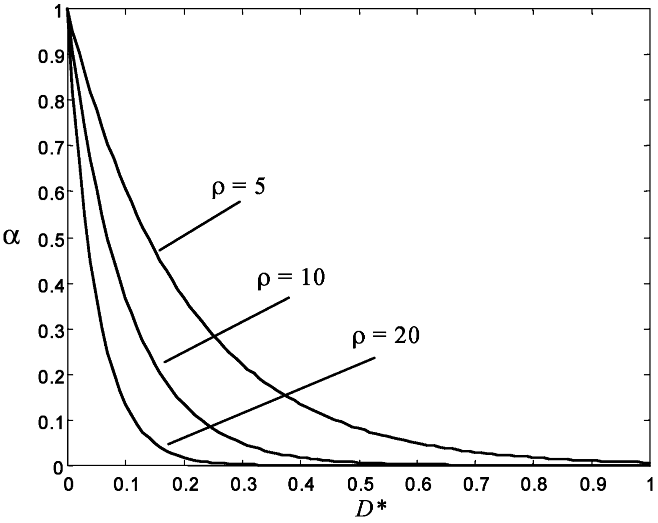

Fig. 15. Trade-off between the normalized antibody affinity D^* and its mutation rate α, according to (Eq. 5), for different values of ρ

where ρ is a parameter that controls the smoothness of the inverse exponential, and D^* is the normalized affinity that can be determined by $D^* = D/D_{\max}$.

Kepler and Perelson (1993) proposed an optimal control treatment for the affinity maturation of an immune response that is different from the one described above. The authors suggested that the maturation of the immune response by somatic hypermutation is marked by a rapid and dramatic increase in affinity for the antigen causing the immune response. An optimal mutation schedule would be the one in which periods of rapid mutation alternate with periods of mutation free growth.

3.2.3 Negative selection

The negative selection of T-cells eliminates those cells whose receptors are capable of recognizing self antigens. This way, all T-cells that survive negative selection are assumed to recognize only nonself antigens. This is a very interesting idea for the development of algorithms that monitor a system against an anomaly or unusual behavior.

Inspired by this idea, Forrest, et al. (1994) developed an anomaly detection algorithm based upon the negative selection of T-cells within the thymus. It was named *negative selection algorithm* and its original application was in computer security. The interesting aspect of this algorithm is that it can be used to perform tasks like pattern recognition by storing information about the set of patterns that are unknown to the system.

The negative selection algorithm is simple and works as follows. Given a self set S of patterns (strings) to be protected, generate a set A of pattern recognizers, named *detectors*, that does not match any string that belong to S. The iterative process of generating the set A can be described as follows and is summarized in Fig. 16.

1. Randomly generate strings and place them in a set P;
2. Determine the affinity of all strings in P with all strings of the self set S;
3. If the affinity of a string of P with at least one string of S is greater than or equal to a given affinity threshold ϵ, then the string in P recognizes the self-string and has to be eliminated (negatively selected); else the string in P belongs to the nonself set and is introduced into the set A.

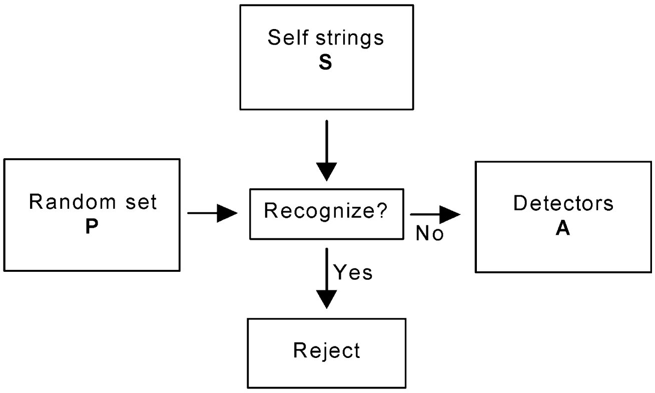

Fig. 16. A negative selection algorithm

3.3 Immune network models

The original immune network theory, proposed by Jerne (1974), suggested an immune system with a dynamic behavior even in the absence of nonself antigens. This proposal was different from clonal and negative selection, as it suggested that B-cells were capable of recognizing each other. This would endow the immune system with a certain type of eigen-behavior and network of communication among cell receptors.

Several theoretical immunologists were interested in creating models of immune networks so as to introduce new ways of explaining how the immune systems works (Perelson, 1989; Farmer et al., 1986). Once researchers in computational intelligence (soft computing) became aware of these works, interest was established in applying these new immune inspired models to solve problems in computing, engineering and other domain areas. The first network models were mainly based on sets of differential equations governing the variations in population sizes of antibody molecules and B-cells. We classify these works as *continuous immune network models*. They have been widely used by the AIS community in applications such as robotics, optimization and control (Ishiguro et al., 1997; Bersini, 1991; Bersini and Varela, 1994). The immune networks also served as inspiration to the development of machine learning network models with applications mainly in data analysis (Timmis, 2000; de Castro and Von Zuben, 2001a). The latter have been classified as *discrete immune network models* as they are not based on differential equations, but iterative procedures of adaptation or difference equations.

The following subsections review a continuous immune network model that has been widely used by researchers on AIS and two discrete immune networks also widely used by researchers in the field.

3.3.1 A continuous immune network model

Farmer et al. (1986) represented immune cells and molecules as bitstrings in a Hamming shape-space, as illustrated in Fig. 17. An antibody molecule was represented by two concatenated portions: one named *epitope* (e) and another named *paratope* (p). The epitope being the portion of the antibody molecule that can be recognized by the paratopes of other antibodies.

Strings were allowed to match complementarily in any possible alignment, modeling the fact that two molecules may react in more than one way. Equation (6) specifies a *matrix of matching* $m_{i,j}$ that corresponds to the degree of matching of each element in the AIS.

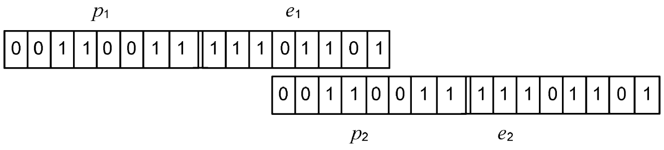

Fig. 17. Bitstrings representing the epitope and paratope of two antibody molecules

$$m_{i,j} = \sum_k G\left(\sum_n e_i(n+k) \wedge p_j(n) - \varepsilon + 1\right) , \qquad (6)$$

where $e_i(n)$ is the n-th bit of the i-th epitope, $p_j(n)$ is the n-th bit of the j-th paratope, $\wedge$ corresponds to the Hamming distance between $e_j(\cdot)$ and $p_j(\cdot)$ and ε corresponds to the affinity threshold. The parameter k corresponds to a given alignment between a paratope and an epitope. If matches occur in more than one alignment, their strengths are summed, including the case of strings with different lengths. The function $G(\cdot)$ measures the strength of a possible reaction between an epitope and a paratope as given by Eq. (7).

$$G(x) = \begin{cases} x & \mathrm{x} > 0 \\ 0 & \text{otherwise} \end{cases} \qquad (7)$$

To quantify the dynamics of the network, it was assumed N antibody types with concentrations $\{c_1,\ldots,c_N\}$ and M antigens with concentrations $\{y_1,\ldots,y_M\}$. The rate of change of antibody concentration is given by

$$\frac{\mathrm{d}c_i}{\mathrm{d}t} = k_1\left[\sum_{j=1}^{N} m_{j,i}c_ic_j - k_2\sum_{j=1}^{N} m_{i,j}c_ic_j + \sum_{j=1}^{M} m_{j,i}c_iy_j\right] - k_3c_i \ , \qquad (8)$$

where the first term represents the stimulation of the paratope of an antibody type i by the epitope of an antibody type j. The second term represents the suppression of antibody of type i when its epitope is recognized by the paratope of type j. The parameter k_1 is a rate constant that depends on the number of collisions per unit time and the rate of antibody production stimulated by a collision. Constant k_2 represents a possible inequality between stimulation and suppression. The third term models the concentration of antigens and the last term models the tendency of cells to die.

Equation (8) controls the dynamics of the network. This is in the sense that antibodies which recognize antigens or other antibodies are amplified in number, whereas antibodies which do not are eliminated. The production of novel antibodies provides the system with the ability to cope with unexpected (or unseen) antigens.

3.3.2 Discrete immune network models

The discrete immune networks differentiate from the continuous models in the sense that their adaptation procedures are not based upon a set of differential equations, but an iterative process of adaptation. This section details the learning algorithms for two discrete immune network models. These were originally developed for pattern recognition, data clustering and data compression. However, it is suggested that these learning algorithms can be considered as generic and can therefore be applied to other domains such as optimization, control and robotics. Each learning algorithm can be used to construct an artificial immune network capable of extracting information from a set of input patterns that corresponds to the antigenic universe. For both algorithms, B-cells and antibodies (Ab) are the main elements of the immune networks, and antigens (Ag) correspond to the input patterns.

3.3.2.1 RAIN

Timmis (2000) proposed an immune network learning algorithm named RAIN (*Resource limited Artificial Immune Network*). Each network element corresponds to a B-cell composed of an antibody (e.g. an attribute string in an Euclidean shape-space), a stimulation level and a record of the number of *resources* held. A resource allocation mechanism is used to control B-cell population and will be discussed later. The network antibodies are initialized by randomly taking a sub-section of the input patterns (Ag), and the stimulation level and record of resources are all initialized with zero.

The next stage is the presentation of the antigenic patterns. Each pattern is presented to each network cell and the stimulation level s_i is determined after presenting all antigens to the cell i, according to Eq. (9).

$$s_i = \sum_{j=1}^{M} (1 - D_{i,j}) + \sum_{k=1}^{n} (1 - D_{i,k}) - \sum_{k=1}^{n} D_{i,k} \tag{9}$$

where M is the number of antigens, n is the number of connected B-cells, $D_{i,j}$ is the Euclidean distance (Eq. 1) between each antigen j and the B-cell i, and $D_{i,k}$ is the Euclidean distance between the cell i and a B-cell k to which it is connected. Note that $(1 - D_{i,j})$ corresponds to the affinity of a B-cell with antigens or other B-cells in the network. In this case, affinity is inversely proportional to distance.

The stimulation level determines which cells are selected for expansion (*clonal expansion*) and which cells are removed from the network (*metadynamics*). In order to decide which cells are to be maintained within the network, a resource allocation mechanism is employed. There are a predefined maximum number of resources in the network, for which each B-cell must compete. Each B-cell is allocated a number of resources in proportion to its stimulation level: the higher the stimulation level, the higher the number of resources allocated. If the number of resources allocated is greater than the maximum number allowed, then the B-cells that hold the least number of resources are removed from the network. This is repeated until the number of resources allocated is less than or equal to the maximum number allowed.

Some of the remaining network cells will be selected for clonal expansion based upon their stimulation level: the higher the stimulation the higher the probability of cloning. Those cells selected for cloning also reproduce in proportion to their stimulation level: the higher the stimulation level the higher the number of clones to be produced.

Affinity maturation allows selected network cells to adapt their antibodies to the antigenic pattern presented. Each antibody is mutated inversely proportional to its B-cell stimulation level: the higher the stimulation the lower the mutation rate. Finally, the mutated clones are matched against all network cells and their affinity is calculated. If their affinity falls below a given threshold, they are linked together.

These processes are repeated until either a fixed number of iterations is performed, or the network reaches a period of stability, i.e. the number of network B-cells remains constant over a given period of time.

The network learning algorithm can be summarized as follows:

1. *Initialization*: create an initial network out of a sub-section of the antigens;
2. *Antigenic presentation*: for each antigenic pattern, do:
    - 2.1. *Clonal selection and network interactions*: for each network cell, determine its stimulation level according to Eq. (9);
    - 2.2. *Metadynamics*: eliminate network cells with low stimulation level via the resource allocation mechanism;
    - 2.3. *Clonal expansion*: select the most stimulated network cells and reproduce them proportionally to their stimulation;
    - 2.4. *Somatic hypermutation*: mutate each clone inversely proportional to its stimulation level;
    - *2.5. Network construction*: select mutated clones to incorporate into the network;
3. *Cycle*: Repeat step 2 until a stopping criterion is met.

3.3.2.2 aiNet

In the immune network learning algorithm proposed by (de Castro and Von Zuben, 2000b), named aiNet (*Artificial Immune Network*), the network is initialized with a small number of elements randomly generated. Each network element corresponds to an antibody molecule, i.e an attribute string represented in an Euclidean shape-space.

The next stage is the presentation of the antigenic patterns. Each pattern is presented to each network cell and their affinity is determined according to Eq. (1).
A number of high affinity antibodies are selected and reproduced (*clonal expansion*) according to their affinity: the higher the affinity, the higher the number of clones to be produced. The clones generated undergo somatic mutation inversely proportional to their antigenic affinity: the higher the affinity, the lower the mutation rate. A number of high affinity clones is selected to be maintained in the network, constituting what is defined as a clonal memory.

The affinity between all remaining antibodies is determined. Those antibodies whose affinity is less than a given threshold are eliminated from the network (*clonal suppression*). All antibodies whose affinity with the antigen is less than a given threshold are also eliminated from the network. Additionally, a number of new randomly generated antibodies are incorporated into the network (*metadynamics*).

The remaining antibodies are incorporated into the network, and their affinity with the existing antibodies is determined. All but one antibody whose affinity is less than a given threshold are eliminated.

The aiNet learning algorithm can be summarized as follows:

1. *Initialization*: create an initial random population of network antibodies;
2. *Antigenic presentation*: for each antigenic pattern, do:

   2.1. *Clonal selection and expansion*: for each network element, determine its affinity with the antigen presented. Select a number of high affinity elements and reproduce (clone) them proportionally to their affinity;

   2.2. *Affinity maturation*: mutate each clone inversely proportional to affinity. Re-select a number of highest affinity clones and place them into a clonal memory set;

   2.3. *Clonal interactions*: determine the network interactions (affinity) among all the elements of the clonal memory set;

   2.4. *Clonal suppression*: eliminate those memory clones whose affinity is less than a pre-specified threshold;

   2.5. *Metadynamics*: eliminate all memory clones whose affinity with the antigen is less than a predefined threshold;

   2.6. *Network construction*: incorporate the remaining clones of the clonal memory with all network antibodies;

   2.7. *Network interactions*: determine the similarity between each pair of network antibodies;

   2.8. *Network suppression*: eliminate all network antibodies whose affinity is less than a pre-specified threshold;
3. *Cycle*: repeat steps 2 to 4 until a pre-specified number of iterations is reached.

Comparison of the discrete immune network models

Although both algorithms may seem rather similar, there are major differences between them in several levels, such as basic network element, immune network interactions, population control mechanism, and interpretation.

In RAIN, the basic element is a B-cell comprised of an antibody attribute string, a stimulation level and a resource allocation indicator, whereas in aiNet the basic element is primarily an antibody attribute string. However, the same way that the stimulation level is part of a B-cell in RAIN, the antibody affinity with antigens and other antibodies could also be viewed as parameters contained within a B-cell in aiNet.

To determine the stimulation level of each network B-cell, RAIN employs a difference equation version of the differential equation proposed by Farmer and collaborators (Eq. 8). This stimulation level takes into account antigenic stimulation and network interactions, thus dictating B-cell survival and reproduction. Similarly, aiNet uses an affinity measure to quantify the degree of antigenic recognition and the degree of interaction with other network antibodies. However, this is performed in different time scales during learning, and not combined into a single equation as in RAIN.

To prevent an exponential growth of the network population, both algorithms employ a population control strategy. In RAIN, a resource allocation mechanism encourages highly stimulated B-cells to survive in the network. This promotes the control of the network size and the creation of a representative internal image of the antigenic universe. In contrast, aiNet attempts to reduce redundancy by eliminating similar antibodies, based upon their degree of similarity (affinity) with other network antibodies. This has the effect of controlling the population size.

Finally, what network results from the learning algorithm? The RAIN learning algorithm produces a topological representation of the antigenic patterns. This allows the identification of important features contained within the antigenic universe, such as clusters and interrelationships between data items. A special tool has been designed to visualize the network structure (Timmis, 2001). In aiNet, a reduced discrete set of antibodies is constructed so as to follow the spatial distribution of the antigenic universe. In order to interpret the resultant aiNet various graph concepts and hierarchical clustering techniques can be utilized (de Castro and Von Zuben, 2001d), such as minimum spanning trees and dendrograms.

As general comments for both learning algorithms, it is important to note that these networks also follow the same structure as the continuous networks. The behavior of the population of network cells is a function of the antigenic and network interactions, added to the metadynamics effects, i.e. influx of new elements and death of unstimulated ones. In addition, although both algorithms were originally implemented using real-valued vectors in an Euclidean shape-space, they are not necessarily restricted to this shape-space.

3.4 An overview of the framework

The framework to design artificial immune systems was inspired by the design methods used in other biologically motivated computing paradigms such as neural networks and evolutionary algorithms. The framework has reviewed and expanded a scheme to create abstract models of the immune cells and molecules. A set of functions can be employed to quantify the interactions of these cells and molecules. Finally, immune algorithms can be employed to govern the dynamics of the AIS, i.e. the behavior along time of the cells and molecules composing the system. Given this framework, the design of an AIS is straightforward.

1. Choose suitable shape-spaces and affinity measures. These are usually problem dependent or determined according to some heuristics.
2. Apply any of the algorithms described (or a new one) to determine how the system is going to behave over time.

Table 1 summarizes all the parts of the framework stressing their main features and rationale.

The four types of immune algorithms (clonal and negative selection, and continuous and discrete network models) have some (but are not restricted to) main domains of application:

- Clonal selection: pattern recognition and optimization;
- Negative selection: anomaly and fault detection;

Table 1. The components of the framework, their main features and rationale

Framework	Main Features	Rationale
Shape-space	A space of attribute strings that correspond to the shapes of cells and molecules	Abstract representation scheme for the immune cells and molecules
Bone marrow models	Processes of generating or concatenating attributes into a string	Generation of cells and molecules in a shape-space
Clonal selection	Evolutionary-like procedure of adaptation in which selection, reproduction and mutation are proportional to affinity	Governs the interactions of the immune cells and molecules when presented with a nonself antigen
Affinity maturation	Mutation process followed by selection and responsible for improving the affinities of cell receptors	Describes how to the elements of the AIS adapt themselves, i.e. alter their attribute strings
Negative selection	Iterative process of comparing (matching) strings	Generates a set of detectors that does not recognize the self antigens
Continuous network	Set(s) of coupled differential equations usually containing terms that account for nonself antigens and the elimination and introduction of new elements	Governs the interaction of immune cells with each other and nonself antigens
Discrete network	Iterative procedure of adaptation that accounts for nonself antigens and the elimination and introduction of new elements	Governs the interactions of immune cells with each other and nonself antigens

- Continuous immune network models: control, robotics, optimization and pattern recognition;
- Discrete immune network models: pattern recognition, data analysis, machine learning and optimization.

4
AIS in context with other SC approaches

The previous section has proposed a simple framework for the design of artificial immune systems. A selected number of existing algorithms (the most widely used ones) with broad application domains were selected to compose this framework. Now that a definition and outline for these AIS has been proposed, it is possible to describe some similarities and differences between artificial immune systems, artificial neural networks, evolutionary algorithms and fuzzy systems. It is assumed that the reader is familiar with the already established soft computing strategies. For good textbooks, please refer to Haykin (1999), Bäck et al. (2000a b), and Pedrycz and Gomide (1998).

The focus of this section will be on the basic elements of the framework:

- *Representation*: what are the main features that characterize an element of a given system and how it differs from the other paradigms;
- *Functions*: what kind of function governs the interactions of the elements of the system with each other and the environment;
- *Algorithms*: which paradigm governs the adaptation of each strategy, such as evolution, learning and rules of inference.

In population-based AIS, attribute strings represent cells and molecules in a shape-space. In network-based AIS, in addition to the attribute strings, there are connections among network cells and molecules, and other parameters, such as affinity with other cells and stimulation level measures. Neural networks have artificial neurons typically composed of an activation function, connection strengths and activation thresholds. These artificial neurons constitute mathematical models of the biological neurons, which perform the inner product of a vector of inputs and the neuron weight vector, and then apply an activation function to this product in order to produce the neuron output. Note that in population-based AIS, the immune cells are basically discrete elements responsible for storing information about the environment. However, in network-based AIS, the immune cells are information processors that determine the affinity with self and nonself antigens (based upon an affinity measure). Evolutionary algorithms are composed of strings representing individual chromosomes. In essence, there is no difference between a chromosome in an EA and an attribute string in a population-based AIS. Fuzzy systems are composed of fuzzy numbers and/or sets characterized by membership functions of linguistic variables mapping elements from one domain, space or universe of discourse, into a unitary interval.

Within artificial immune systems, there are basically two types of functions employed to quantify the interactions of individual cells and molecules: fitness and affinity functions. Fitness functions are used when the quality of the elements of the AIS is evaluated in tasks that do not involve comparing them with other elements, in this latter case an affinity function is used. It means that, for example, in pattern recognition applications where an immune cell is compared with another attribute string in order to evaluate their degree of interaction (similarity or difference), an affinity measure (such as Eqs. (1)–(3)) is used. Evolutionary algorithms typically have a fitness function that evaluates the quality of each individual of the population in relation to the environment. The structure of a fuzzy system is in most cases based upon a discrete set of membership functions for the linguistic variables, and fuzzy rules that determine the relations of the variables. A set of norms is used to compute with the fuzzy rules and numbers.

Adaptation in artificial immune systems may involve different paradigms such as learning and/or evolution. Population-based AIS usually have an evolutionary-like type of adaptation. Immune network models by contrast, might present a mixture of evolution and (un)supervised learning in their adaptation algorithms. Most neural networks have a learning algorithm (e.g. Hebb rule or backpropagation of errors) or rule (e.g. pseudo-inverse method in discrete Hopfield (1984) networks) falling into one of three major paradigms: supervised, unsupervised and reinforcement learning. In fuzzy reasoning, the fuzzy rules fired by a given input stimulus are aggregated by a compositional rule in order to infer an output. The environment might be represented by a crisp or a fuzzy number that is going to activate (fire) one or more fuzzy rule.

5
A survey of hybrid models

Soft computing is mainly concerned with the integration of computational intelligence paradigms in order to create hybrids with the benefit of combining different paradigms. After introducing the field of artificial immune systems and discussing similarities and differences between AIS and the other approaches, the focus now turns into a survey of the works from the literature that propose hybrids of AIS with artificial neural networks (ANN), evolutionary algorithms (EA) and fuzzy systems (FS). For a broader theoretical comparison of these paradigms with AIS, please refer to de Castro and Timmis, (2002a).

Generally speaking, ANN, EA and FS have a great potential to interact with artificial immune systems. The majority of the works that are described in the literature trying to integrate one or more of these soft computing paradigms with AIS, involve artificial neural and immune network models, or immune and evolutionary algorithms. The review focuses on how the strategies benefit from the integration.

5.1
Artificial immune systems and artificial neural networks

Trades-off between the immune and the nervous systems date back from the early days of theoretical immunology. However, the proposal of the immune network theory by Jerne in 1974 stimulated several researchers to look into both systems and to try and trace new parallels between them. As a natural outcome of this interdisciplinary research, immune network models have been used as novel approaches for the development and improvement of neural network models and vice-versa.

In Dasgupta (1997), the author performed one of the first attempts at comparing AIS with ANN models. However, the main focus of this work was on the biological nervous and immune systems. A further attempt was undertaken by de Castro and Von Zuben (2001a) where the authors compared their artificial immune network model (aiNet) with ANNs, focusing self-organizing neural networks.

Hoffmann (1986) and Hoffmann et al. (1986) used the analogy between the immune network theory and the central nervous system to formulate a neural network model. The immune system was viewed as an L-dimensional system containing a very large number of singular points representing attractors. Learning in this system corresponded to altering the strength of the stimuli being presented to the network, with the connection strengths among cells being kept fixed. This approach is in contrast with the traditional artificial neural networks, in which the weight vectors are adapted to the input data.

The works of Vertosick and Kelly (1989, 1991) proposed that the immune system might represent an alternative paradigm in which to search for neural network architectures. Based on the Parallel Distributed Processing (PDP) theory (Rumelhart et al., 1986), they tried to map the immune network theory into a PDP immune network. No learning algorithm was explicitly presented, though the authors strongly suggested that the learning behavior of the immune system is unsupervised.

Based on the immune metadynamics, i.e. the capability of recruiting new cells and molecules into the system and disposing the not useful ones, Abbattista et al. (1996) developed a discrete associative network. This mechanism was used to define a population of points in a Hamming shape-space. The best points of this population were taken as the attractors representing the network memories. As in the discrete Hopfield network case, this algorithm was composed of a learning (storing) phase and a recall phase.

In de Castro and Von Zuben (2002), the authors developed a growing Boolean competitive network based on the clonal selection and affinity maturation principles of the immune system. The main features of the proposed algorithm are competitive learning, automatic generation of the network with growing and pruning phases, and binary representation of the connection strengths in a Hamming shape-space. The weight updating procedure is a guided mutation search that simulates the affinity maturation process of the antibody repertoire, such that the weights (antibodies) become a more perfect complement of the antigens to be recognized.

An approach to develop a simulated annealing algorithm (Kirkpatrick et al., 1987) based on the immune metaphor was proposed by de Castro and Von Zuben (2003). This was applied to the problem of initializing multilayer feedforward neural networks trained using an error backpropagation algorithm. The authors argued that the correlation between the quality of the initial network weights and the quality of the network output could be likened to the quality of the initial antibody repertoire and the quality of the immune response. The authors extracted the metaphors of creating antibody diversity using the idea of an Euclidean shape-space. They proposed an algorithm capable of generating a set of initial weight vectors diverse enough to reduce the likelihood of the feedforward neural network to converge to a local optimum.

The immune network model called aiNet described previously was used by de Castro and Von Zuben (2001c) to implement an unsupervised approach to determine the number and position of radial basis functions to be used in RBF neural networks. The main goals of the algorithm were to cluster and filter unlabeled numerical data sets. The authors employed an Euclidean shape-space to represent the molecules. Here an antibody corresponded to a candidate center for the RBF neural network and an antigen was equivalent to an input pattern.

5.2 Artificial immune systems and evolutionary algorithms

A great number of the AIS currently developed can be characterized as having an adaptation akin to an evolutionary algorithm. This section describes those works that explicitly take into account an evolutionary algorithm as part of its processing or which combines it with an AIS in order to improve individual performances. The focus will be given to artificial immune systems that account for the formation of niches, species and diverse populations (basically those applied to multimodal function optimization), and to the ones that integrate with a genetic algorithm (GA) or a genetic programming (GP) approach.

A binary immune system model was used by Forrest et al. (1993) in order to study pattern recognition and learning in artificial immune systems. A genetic algorithm was used to study the maintenance of diversity and generalization capability of an AIS. In this case, generalization means the detection of common schemas that are shared among many antigens. Population diversity in contrast, corresponds to a set of individuals capable of broadly covering the affinity landscape. The authors used a simple binary Hamming shape-space to represent the molecules and the Hamming distance as the affinity measure.

Hightower et al. (1995) studied the effects of evolution on the genetic encoding for antibody molecules and the application of an evolutionary algorithm to the production of initial repertoires of cells and molecules for AIS. They used a bone marrow model similar to the one described in Sect. 3.1. For further works in this area see Perelson et al. (1996), and Oprea and Forrest (1999).

Potter and de Jong (1998) presented an approach for concept learning in which a co-evolutionary genetic algorithm was used to construct an artificial immune system whose antibodies were capable of discriminating examples from counter examples, i.e. self from nonself. They explored the generality and diversity controlling mechanisms of their AIS.

Hajela and Yoo (1999) proposed that the immune system capabilities of performing pattern (schema) recognition and adaptation could be used advantageously to improve the performance of genetic algorithms in structural optimization problems. Their work focused on two aspects: using the immune system capabilities to enhance the convergence of a GA approach, and handling the design of constraints in the GA-based optimization.

Hart and Ross (1999) investigated whether an AIS could be evolved using a genetic algorithm and then be used to produce sets of schedules, which collectively cover a range of contingencies, both predictable and unpredictable. Their model included evolution through gene libraries, affinity maturation of the immune response and the clonal selection principle.

Dasgupta et al. (1999) proposed what they called an immunogenetic approach to recognize spectra for chemical analysis. In their approach, a standard genetic algorithm was used to create a library of specialists to perform the central administration of spectrum recognition.

Nikolaev et al. (1999) introduced an immune version of Genetic Programming (GP). In this immune-GP version (iGP), the progressive search was controlled by a dynamic fitness function, based on an analogy with an immune network model. The programs were reinforced with rewards for matching important examples and stimulated to match different examples. The fitness function consisted of two dynamic models that exerted influence on each other: (1) a model for propagating programs that match more important examples, and (2) a model for changing the importance of examples in relation to the number of programs that recognize it.

5.3 Artificial immune systems and fuzzy systems

In the immune system, antigenic recognition is approximate, i.e. an immune response can be elicited even when the binding between an antigen and an antibody is not perfect; an approximate binding might suffice. Together with cross-reactivity, these characteristics stress the presence of “fuzzyness” within the immune systems, suggesting that fuzzy logic might be appropriate to model several aspects and mechanisms of the immune system. Consequently, fuzzy systems and artificial immune systems may provide fruitful interactions; will be reviewed in the following examples.

Krishnakumar et al. (1995) proposed a computational system employing immune metaphors and other soft computing techniques. The proposed hybrid system was composed of combinations of artificial immune systems with artificial neural networks, fuzzy systems and evolutionary algorithms, according to the problem under study. This approach was evaluated in a control application. In this particular case, several elements of the immune system were equated to different levels of control strategies. As examples, the innate immunity was likened to a robust feedback controller and B-cell activation was compared to a process capable of modifying parameters of the controller. Immune processes such as affinity maturation were stressed as important for control, but the approach focused on parallels between the immune system and the control problem.

Baldwin (1896) suggested that characteristics learnt or acquired during the lifetime of an individual could become part of the genetic makeup of succeeding generations without Lamarckian inheritance. He argued that the learning of the acquired useful characteristics increase individual survival probabilities, even if these characteristics are not genetically transmitted. Hightower et al. (1996) used a non-linear (sigmoid) binding function to determine the binding value between two molecules. The authors argued that a non-linear function is more plausible from an immunological perspective and a necessary requirement for the Baldwin effect to occur while evolving a binary immune system model using an evolutionary algorithm.

In the works of Lee et al. (1999) and Jun et al. (1999), the authors used different types of antibodies, each with a specific task. In those papers, a stimulation level of a given antibody was a function of its percentage of success in the execution of a given task, based on a function similar to the one presented in Fig. 18. This function is typical in fuzzy systems, where the universe of discourse of a

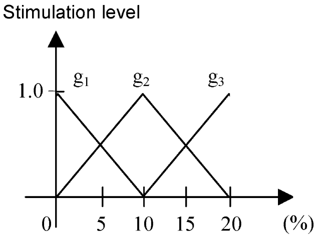

Fig. 18. Function that determines the stimulation level of a given antibody

variable (in this case the percentage of success) is partitioned into several intervals, leading to an approximate representation of the variable under study.

In de Castro and Von Zuben (2001d), the authors proposed an artificial immune network model to perform data analysis, in particular data clustering. In this paper, the representation of clusters by their centroids allowed them to assign membership levels to each immune network cell in relation to the determined clusters, yielding a fuzzy clustering scheme. This scheme extended the notion that each network cell belongs to a single cluster and associates each cell with every network cluster using a fuzzy membership function.

6 Trends on the creation of novel hybrid models

The practical application of most of the soft computing paradigms discussed so far requires the specification of some features of the algorithms such as *model selection* and *adaptation parameters*.

Model selection corresponds to the definition of a suitable structure for the system. In neural networks, it involves basically the definition of the number of connections, neurons and layers to be used in the network. Evolutionary algorithms usually have a fixed number of individuals (chromosomes) in the population that has to be defined a priori. In fuzzy systems, the number of partitions for each linguistic variable and the fuzzy rules also have to be set up a priori.

Another important aspect that affects most of the soft computing approaches, is the necessity to a priori define some *adaptation parameters*. In neural networks, parameters like learning rates, momentum terms, activation functions, and neighborhood decaying rates, have to be defined according to the network type and/or learning algorithm. In the case of evolutionary algorithms, genetic variation probabilities, selection and reproduction strategies are part of the user-defined parameters. In fuzzy systems, different types of membership functions (Gaussian, triangular, trapezoidal, etc.) lead to different performances of the algorithm. In addition, several types of t-norms and s-norms are available for use, and the choice is in most cases dependent upon the problem under study or the designers' preference or expertise.

6.1 What has already been done

The survey presented in the previous section shows that some of these problems have already been tackled by the hybrids of AIS with neural networks, evolutionary algorithms and fuzzy systems. Based on this survey, Table 2 summarizes the main outcomes already resulted from the combination of AIS with each of these approaches separately.

6.2 Hints on what can still be done

Last but not least it is possible highlight some future avenues for the integration of AIS with the other soft computing paradigms.

Evolutionary algorithms have been widely used to model and parameter selection for other approaches. For example, they can be used to evolve artificial neural network architectures (Harrald and Kamstra, 1997; Maniezzo, 1994; Friedrich and Moraga, 1996; Opitz and Shavlik, 1997) or to design fuzzy systems (Chan et al., 1997; Shi et al., 1999; Belarbi and Titel, 2000). Neurofuzzy systems, i.e. the combination of ANN with fuzzy systems, usually have the advantage of allowing an easy translation of the final system into a set of if-then rules, and the fuzzy system can be viewed as a neural network structure with knowledge distributed throughout connection strengths (Kosko, 1992). Several examples of neurofuzzy systems can be found in the literature (e.g. Kuo et al., 1993; Kwan and Cai, 1994; Chiang and Gader, 1997; Juang, 2000; Pal et al., 2000). More complex hybrids, combining fuzzy, neural and evolutionary algorithms are also available in the literature (e.g. Krishnakumar et al., 1995; Iyoda et al., 1999).

All these works suggest that evolutionary algorithms can also be used to search for an adequate model selection

Table 2. Integrative benefits shared by AIS, ANN, EA and FS

Integration	Outcome
AIS ↔ ANN	• AIS have suggested new ANN models, architectures and learning algorithms • AIS provided increased memory capacities for ANN • AIS were used to develop new intialization techniques for ANN
AIS ↔ EA	• EAs provide new definition of intial repertories for AIS • EAs have been used to study the evalution of the genetic encoding of AIS • AIS were used to enhance GA convergence • AIS have been used to handle constraints in GAs • AIS were used to develop co-evolutionary GAs • AIS have been used to promote and maintain niches, species and diversity in evolutionary algorithms • An immune version of genetic programming was proposed
AIS ↔ FS	• Fuzzy logic has been used to model approximate binding in AIS • FS lead to more biologicaly appealing AIS algorithms • A fuzzy binding was used to simulate the Baldwin effect in AIS • AIS have provided alternative fuzzy clustering schemes • AIS can be used to model selection in FS

and automatic parameter definition for artificial immune systems. Also, the integration of fuzzy logic with AIS may lead to hybrid systems that are more biologically plausible, that can be expressed in the form of a set of if-then rules, and that can compute with fuzzy or incomplete information. Neural networks can provide alternative learning algorithms, network architectures, types of cells and nonlinearities for immune networks, and vice-versa.

The problem of model selection in ANN can be dealt with several approaches, among which constructive (Kwok and Yeung, 1997; Fritzke, 1994) and pruning strategies (Reed, 1993; de Castro and Von Zuben, 1999) are the most common. Evolutionary algorithms with adaptive population sizes can also be found in the literature (e.g. Arabas et al., 1994; Krink et al., 1999). Automatic methods to define the number of partitions and their respective positions for linguistic variables in fuzzy systems were suggested in Caminhas et al. (1995). Strategies to automatically determine user-defined parameters have also been implemented in all the SC approaches (e.g. Yu and Chen, 1997; Lobo, 2000; Angeline, 1995). It is possible that all these strategies applied to other SC approaches may shed some light into the solution of the respective problems in AIS: model selection and automatic determination of adaptation parameters.

As one last question to be raised 'How equivalent are AIS with other computational intelligence and search methods?, This is an interesting and intriguing matter that requires exploration far beyond the scope of this paper. Work can be found in the literature that draws parallels between evolutionary and gradient-based search strategies (e.g. Salomon, 1998) and fuzzy logic systems with specific neural network architectures (e.g. Jang and Sun, 1993; Hunt et al., 1996; Li and Chen, 2000), suggesting that similar equivalencies could be made between AIS and these paradigms. This is still an open question even for the more established fields of research.

7
Discussion

Artificial immune systems constitute a novel computational intelligence paradigm inspired by the immune system. Like neural networks and evolutionary algorithms, AIS are highly abstract models of their biological counterparts applied to solve problems in different domain areas. AIS have also been used in conjunction with other soft computing paradigms in order to create more powerful models and improve individual performances, supporting the claim that they compose a new and very useful soft computing approach.

This paper explored several aspects of artificial immune systems. It first presented a brief introduction to the vertebrate immune system in order to provide the reader with the necessary biological background to understand, develop, implement and hybridize AIS. The second main contribution of this paper was the proposal of a general framework for designing AIS. To develop this framework, we were inspired by the basic design principles of other soft computing approaches motivated by biology, such as neural networks and evolutionary algorithms. Under this perspective, the framework is conceptually simple: it is composed of a formal methodology to represent the components of the system, a set of functions that evaluate the quality of each of these components in a given environment, and a set of algorithms that govern the overall behavior of the system. Each of these three layers of the framework was described, and the role played by all of them summarized in a single section. It is hoped that this framework sheds some light into the development and understanding of AIS.

The paper followed with a conceptual comparison of AIS with artificial neural networks, evolutionary algorithms and fuzzy systems. This comparison was made based upon the framework proposed, i.e. what is the basic representation, functions and algorithms involved in each paradigm, and how they differ from each other. There then followed a survey of works from the literature that hybridize AIS with all the other approaches, and a discussion of the benefits of this integration was provided. It was argued that more useful and powerful algorithms have already arisen and can still arise when two or more of the different paradigms are hybridized. The main results of integrating the already established soft computing techniques (ANN, EA and FS) were reviewed, and it was suggested that similar hybridization with AIS could be performed.

References

Ada GL, Nossal G (1987) The clonal selection theory, Scientific American **257**(2): 50–57

Abbattista F, Di Gioia G, Di Santo G, Farinelli M (1996) An associative memory based on the immune networks, 2nd Online Workshop on Evolutionary Computation, Nagoya, Japan

Angeline PJ (1995) Adaptive and self-adaptive evolutionary computations. In: Computational Intelligence: A Dynamic Systems Perspective, Palaniswami M, Attikiouzel Y, Marks R, Fogel D, Fukuda T (eds), Piscataway, NJ:IEEE Press, pp. 152–163

Arabas J, Michalewicz Z, Mulawka J (1994) GAVaPS – A genetic algorithm with varying population size, Proc of the Conference on Evolutionary Computation (CEC'94), pp. 73–78

Bäck T, Fogel DB, Michalewicz Z (2000a) Evolutionary Computation 1 Basic Algorithms and Operators, Institute of Physics Publishing (IOP), Bristol and Philadelphia

Bäck T, Fogel DB, Michalewicz Z (2000b) Evolutionary Computation 2 Advanced Algorithms and Operators, Institute of Physiscs Publishing (IOP), Bristol and Philadelphia

Baldwin JM (1896) A new factor in evolution, American Naturalist **30**: 441–451

Belarbi K, Titel F (2000) Genetic algorithm for the design of a class of fuzzy controllers: an alternative approach, IEEE Trans Fuzzy Systems **8**(4): 398–405

Berek C, Ziegner M (1993) The maturation of the immune response, Imm Today **14**(8): 400–402

Bersini H (1991) Immune Network and Adaptive Control, Proc of the First European Conference on Artificial Life, MIT Press, pp. 217–226

Bersini H, Varela FJ (1994) The immune learning mechanisms: reinforcement, recruitment and their applications. In: Computing with Biological Metaphors, Paton R (ed.), Chapman & Hall

Bonissone PP (1997) Soft computing: The convergence of emerging reasoning technologies, Soft Computing **1**(1): 6–18

Burnet FM (1959) The Clonal Selection Theory of Acquired Immunity, Cambridge University Press

Caminhas WM, Tavares HMF, Gomide F (1995) Competitive learning of fuzzy, logical neural networks, Proc of the VI IFSA World Congress **2**: 639–642

Carter JH (2000) The immune system as a model for pattern recognition and classification, J Amer Medical Informatics Assoc **7**(1)

de Castro LN, Timmis J (2002a) Artificial immune Systems: A New Computational Intelligence Paradigm, Springer-Verlag

de Castro LN, Timmis J (2002b) Artificial immune systems: a novel paradigm for pattern recognition. In: Artificial Neural Networks in Pattern Recognition, Alonso L, Corchado J, Fyfe C (eds), pp. 67–84. University of Paisley

de Castro LN, Von Zuben FJ (1999) An improving pruning technique with restart for the Kohonen self-organizing feature map, Proc Int Joint Conf Neural Networks **3**: 1916–1919

de Castro LN, Von Zuben FJ (2000a) The clonal selection algorithm with engineering applications, Proc of GECCO'00, Workshop Proceedings, pp. 36–37

de Castro LN, Von Zuben FJ (2000b) An evolutionary immune network for data clustering, Proc of the IEEE Brazilian Symposium on Neural Networks, pp. 84–89

de Castro LN, Von Zuben FJ (2000c) Artificial immune systems: Part II – A survey of applications, Technical Report – RT DCA 02/00, p. 65. [On line] www.dca.fee.unicamp.br/∼lnunes

de Castro LN, Von Zuben FJ (2001a) Immune and neural network models: theoretical and empirical comparisons, Int Journal of Computer Intelligence and Applications **1**(3): 239–257

de Castro LN, Von Zuben FJ (2001b) An immunological approach to initialize feedforward neural network weights, Proc of ICANNGA'01, pp. 126–129

de Castro LN, Von Zuben FJ (2001c) Automatic determination of radial basis function: an immunity-based approach, Int J Neural Systems (Special Issue on Non-Gradient Learning Techniques), **11**(6): 523–535

de Castro LN, Von Zuben FJ (2001d) aiNet: an artificial immune network for data analysis. In Data Mining: A Heuristic Approach, Abbass HA, Sarker RA, Newton CS (eds), Idea Group Publishing, USA, Chapter XII, pp. 231–259

de Castro LN, Von Zuben (2003) The construction of a boolean competitive network using ideas from immunology, Neurocomputing, **50C**: 51–85

Chan KCC, Lee V, Leung H (1997) Generating fuzzy rules for target tracking using a steady-state genetic algorithm, IEEE Trans on Evolut Comput **1**(3): 165–178

Chiang J-H, Gader PD (1997) Hybrid fuzzy-neural system in handwritten word recognition, IEEE Trans on Fuzzy Systems, **5**(4): 497–510

Dasgupta D (ed) (1999) Artificial Immune Systems and Their Applications, Springer-Verlag

Dasgupta D, Cao Y, Yang C (1999) An immunogenetic approach to spectra recognition, Proc of GECCO'99, pp. 149–155

Dasgupta D (1997) Artificial neural networks and artificial immune systems: similarities and differences, Proc of the IEEE SMC **1**: 873–878

Dasgupta D, Forrest S (1996) Novelty detection in time series data using ideas from immunology, Proc of the ISCA'96

Farmer JD, Packard NH, Perelson AS (1986) The immune system, adaptation, and machine learning, Physica 22D: 187–204

Forrest S, Perelson A, Allen L, Cherukuri R (1994) Self-nonself discrimination in a computer, Proc of the IEEE Symposium on Research in Security and Privacy, 202–212

Forrest S, Javornik B, Smith RE, Perelson AS (1993) Using genetic algorithms to explore pattern recognition in the immune system, Evolut Comput **1**(3): 191–211

Forrest S, Perelson A (1992) Computation and the immune system, SIGBIO Newsletter, Association for Computing Machinery **12**(2): 52–57

Friedrich CM, Moraga C (1996) An evolutionary method to find good building-blocks for architectures of artificial neural networks, Proc of the IPMU'96, 951–956

Fritzke B (1994) Growing cell structures – a self-organizing network for unsupervised and supervised learning, Neural Networks **7**(9): 1441–1460

Fukuda T, Mori K, Tsukiama M (1998) Parallel search for multimodal function optimization with diversity and learning of immune algorithm. In: Artificial Immune Systems and Their Applications, Dasgupta D (ed), Springer-Verlag, pp.210–220

Hajela P, Lee J (1996) Constrained genetic search via schema adaptation: an immune network solution, Structural Optimization **12**(1): 11–15

Hajela P, Yoo JS (1999) Immune network modelling in design optimization. In: New Ideas in Optimization, Corne D, Dorigo M, Glover F (eds), McGraw Hill, London, pp. 203–215

Harrald PG, Kamstra M (1997) Evolving artificial neural networks to combine financial forecasts, IEEE Trans Evolut Comput **1**(1): 40–52

Hart E, Ross P (1999) The evolution and analysis of a potential antibody library for use in job-shop scheduling. In: New Ideas in Optimization, Corne D, Dorigo M, Glover F (eds), McGraw Hill, London, pp. 185–202

Haykin S (1999) Neural Networks – A Comprehensive Foundation, 2nd edn, Prentice Hall,

Hightower RR, Forrest S, Perelson AS (1996) The baldwin effect in the immune system: learning by somatic hypermutation. In: Belew RK, Mitchel M (eds), Adaptive Individuals in Evolving Populations, Addison-Wesley, Reading, MA, pp. 159–167

Hightower RR, Forrest SA, Perelson AS (1995) The evolution of emergent organization in immune system gene libraries. In: Proc of the Sixth ICGA, Eshelman LJ (ed), Morgan Kaufmann, San Francisco, CA, pp. 344–350

Hodgkin PD (1998) Role of cross-reactivity in the develpment of antibody responses, The Immunologist, **6**(6): 223–226

Hoffmann GW (1986) A neural network model based on the analogy with the immune system, J Theor Biol **122**: 33–67

Hoffmann GW, Benson MW, Bree GM, Kinahan PE (1986) A teachable neural network based on an unorthodox neuron, Physica 22D, 233–246

Hofmeyr SA (2000) An interpretative introduction to the immune system. In: Design Principles for the Immune System and Other Distributed Autonomous Systems, Cohen I, Segel LA (eds), Oxford University Press

Hopfield JJ (1984) Neurons with graded response have collective computational properties like those of two-state neurons, Proc Natl Acad Sci USA **81**: 3088–3092

Hunt JE, Cooke DE (1996) Learning using an artificial immune system, J Network Comp Applications **19**: 189–212

Hunt KJ, Haas R, Murray-Smith R (1996) Extending the functional equivalence of radial basis function networks and fuzzy inference systems, IEEE Trans on Neural Networks **7**(3): 776–781

Ishida Y, Hirayama H, Fujita H, Ishiguro A, Mori K (eds) (1998) Immunity-Based Systems – Intelligent Systems by Artificial Immune Systems, Corona Pub. Co. Japan (in Japanese)

Ishiguro A, Watanabe Y, Kondo T (1997) A robot with a decentralized consensus-making mechanism based on the immune system, Proc ISADS'97, pp. 231–237

Iyoda EM, de Castro LN, Gomide FAC, Von Zuben FJ (1999) Evolutionary design of neuro-fuzzy networks for pattern classification, Proc of CEC'99, **2**: 1237–1244

Jang JSR, Sun CT (1993) Functional equivalence between radial basis function networks and fuzzy inference systems, IEEE Trans, on Neural Networks **4**(1): 156–158

Jerne NK (1974) Towards a network theory of the immune system, Ann Immunol (Inst. Pasteur) **125C**: 373–389

Juang J-G (2000) Fuzzy neural network approaches for robot gait synthesis, IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics **30**(4): 594–600

Jun J-H, Lee D-W, Sim K-B (1999) Realization of cooperative and swarm behavior in distributed autonomous robotic systems using artificial immune system, Proc of the IEEE SMC'99 **4**: 614–619

Kephart JO (1994) A biologically inspired immune system for computers. In: Artificial Life IV Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems, Brooks RA, Maes P (eds) MIT Press, pp. 130–139

Kepler TB, Perelson AS (1993) Somatic hypermutation in B cells: An optimal control treatment, J Theor Biol **164**: 37–64

Kim J, Bentley P (1999) The human immune system and network intrusion detection, Proc of the EUFIT'99, CD ROM

Kirkpatrick S, Gelatt Jr CD, Vecchi MP (1987) Optimization by simulated annealing, Science **220**(4598): 671–680

Kosko B (1992) Neural Networks and Fuzzy Systems – A Dynamical Systems Approach to Machine Intelligence, University of Southern California, Prentice-Hall International, Inc

Krink T, Mayoh BH, Michalewicz Z (1999) A patchwork model for evolutionary algorithms with structured and variable size populations, Proc of GECCO **2**: 1321–1328

KrishnaKumar K, Satyadas A, Neidhoefer J (1995) An immune system framework for integrating computational intelligence paradigms with applications to adaptive control. In: Computational Intelligence A Dynamic System Perspective, Palaniswami M, Attikiouzel Y, Marks II RJ, Fogel D, Fukuda T (eds), IEEE Press, pp. 32–45

Kuo Y-H, Kao C-I, Chen J-J (1993) A fuzzy neural network model and its hardware implementation, IEEE Trans. on Fuzzy Systems **1**(3): 171–183

Kwan HK, Cai Y (1994) A fuzzy neural network and its application to pattern recognition, IEEE Trans on Fuzzy Systems **2**(3): 184–193

Kwok TY, Yeung DY (1997) Constructive algorithms for structure learning in feedforward neural networks for regression problems, IEEE Trans on Neural Networks **8**(3): 630–645

Langman RE, Cohn M (1986) The 'Complete' Idiotype Network is an Absurd Immune System, Imm Today **7**(4): 100–101

Lee D-W, Jun H-B, Sim K-B (1999) Artificial immune system for realization of cooperative strategies and group behavior in collective autonomous mobile robots, Proc of AROB'99, 232–235

Li H-X, Chen CLP (2000) The equivalence between fuzzy logic systems and feedforward neural networks, IEEE Trans on Neural Networks **11**(2): 356–365

Lobo F (2000) The Parameter-less Genetic Algorithm: Rational and Automated Parameter Simplified Genetic Algorithm Operation, Dissertation from the University of Lisbon, Portugal

Maniezzo V (1994) Genetic evolution of the topology and weight distribution of neural networks, IEEE Trans on Neural Networks **5**(1): 39–53

Mannie MD (1999) Immunological self/nonself discrimination, Immuno Res **19**(1): 65–87

Nikolaev NI, Iba H, Slavov V (1999) Inductive genetic programming with immune network dynamics, Adv Genetic Programming 3, MIT Press, pp. 355–376

Nossal GJV (1994) Negative selection of lymphocytes, Cell **76**: 229–239

Novák V (1998) Towards formal theory of soft computing, Soft Computing **2**(1): 4–6

Opitz DW, Shavlik JW (1997) Connectionist theory refinement: genetically searching the space of network topologies, J Artificial Intelligence Res **6**: 177–209

Oprea M (1999) Antibody Repertoires and Pathogen Recognition: The Role of Germline Diversity and Somatic Hypermutation, Ph.D. Dissertation, University of New Mexico, Albuquerque, New Mexico, EUA

Oprea M, Forrest S (1999) How the immune system generates diversity: pathogen space coverage with random and evolved antibody libraries, Proc of GECCO'99 **2**: 1651–1656

Pal SK, De RK, Basak J (2000) Unsupervised feature evaluation: a neuro-fuzzy approach, IEEE Trans on Neural Networks **11**(2): 366–376

Pedrycz W, Gomide FAC (1998) An Introduction to Fuzzy Sets Analysis and Design, MIT Press

Perelson AS (1989) Immune network theory, Imm Rev **110**: 5–36

Perelson AS, Hightower R, Forrest S (1996) Evolution and somatic learning in V-region genes, Res Immunol **147**: 202–208

Perelson AS, Oster GF (1979) Theoretical studies of clonal selection: minimal antibody repertoire size and reliability of self-nonself discrimination, J Theor Biol 81: 645–670

Potter MA, de Jong KA (1998) The coevolution of antibodies for concept learning, Proc. of the 5th ICPPSN, pp. 530–539

Rechenberg I (1994) Evolution strategy. In: Computational Intelligence Imitating Life, Zurada JM, Marks II RJ, Robinson CJ (eds), IEEE Press, pp. 147–159

Reed R (1993) Pruning algorithms – a survey, IEEE Trans. on Neural Networks **4**(5): 740–747

Rumelhart DE, McClelland JL, The PDP Research Group (eds) (1986) Parallel Distributed Processing, Cambridge MIT Press

Salomon R (1998) Evolutionary algorithms and gradient search: similarities and differences, IEEE Trans Evolut Comput **2**(2): 45–55

Shi Y, Eberhart R, Chen Y (1999) Implementation of Evolutionary Fuzzy Systems, IEEE Trans on Fuzzy Systems **7**(2): 109–119

Smith DJ (1997) The Cross-Reactive Immune Response: Analysis, Modelling and Application to Vaccine Design, Ph.D. Dissertation, Computer Science, The University of New Mexico, Albuquerque, New Mexico, USA

Timmis J (2000) Artificial Immune Systems: A Novel Data Analysis Technique Inspired by the Immune Network Theory, Ph.D. Dissertation, Department of Computer Science, University of Wales

Timmis J (2001) aiVIS: Artificial Immune Network Visualisation. In: EuroGraphics UK 2001 Conference Proceedings, pp. 61–69, University College London

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

Tonegawa S (1983) Somatic generation of antibody diversity, Nature **302**: 575–581

Vertosick FT, Kelly RH (1991) The immune system as a neural network: A multi-epitope approach, J Theor Biol **150**: 225–237

Vertosick FT, Kelly RH (1989) Immune network theory: A role for parallel distributed processing? Immunology, **66**: 1–7

Yu X-H, Chen G-A (1997) Efficient backpropagation learning using optimal learning rate and momentum, Neural Networks **10**(3): 517–528

Zadeh LA (1997) What is soft computing, Soft Computing **1**(1)