Available online at www.sciencedirect.com

ScienceDirect

Theoretical Computer Science 403 (2008) 11–32

Theoretical Computer Science

www.elsevier.com/locate/tcs

Theoretical advances in artificial immune systems

J. Timmis[a,b], A. Hone[c,*], T. Stibor[d], E. Clark[a]

[a] *Department of Computer Science, University of York, UK*
[b] *Department of Electronics, University of York, UK*
[c] *Institute of Mathematics, Statistics and Actuarial Science, University of Kent, Canterbury, UK*
[d] *Department of Computer Science, Darmstadt University of Technology, Germany*

Received 6 June 2007; received in revised form 21 January 2008; accepted 1 February 2008

Communicated by G. Rozenberg

Abstract

Artificial immune systems (AIS) constitute a relatively new area of bio-inspired computing. Biological models of the natural immune system, in particular the theories of clonal selection, immune networks and negative selection, have provided the inspiration for AIS algorithms. Moreover, such algorithms have been successfully employed in a wide variety of different application areas. However, despite these practical successes, until recently there has been a dearth of theory to justify their use. In this paper, the existing theoretical work on AIS is reviewed. After the presentation of a simple example of each of the three main types of AIS algorithm (that is, clonal selection, immune network and negative selection algorithms respectively), details of the theoretical analysis for each of these types are given. Some of the future challenges in this area are also highlighted.

Keywords: Artificial immune systems; Clonal selection; Negative selection; Immune networks; Markov chains; *k*-CNF satisfiability

1. Introduction

This paper reviews the current theoretical advances in the area of Artificial Immune Systems (AIS). AIS have been defined in [1] as

> "Adaptive systems, inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving."

They are one among many types of algorithm inspired by biological systems, including evolutionary algorithms, swarm intelligence, neural networks and membrane computing. AIS are bio-inspired algorithms that take their inspiration from the human immune system. Within AIS, there are many different types of algorithm, and research to date has focussed primarily on the theories of immune networks, clonal selection and negative selection. These theories have been abstracted into various algorithms and applied to a wide variety of application areas. However, as

[*] Corresponding author. Tel.: +44 0 1227 827397; fax: +44 0 1227 827932.
E-mail address: anwh@kent.ac.uk (A. Hone).

0304-3975/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2008.02.011

noted in [2], a somewhat "scattergun" approach has been adopted, without careful consideration of whether such immune-inspired approaches are actually suitable for the application areas in question. Whilst there has been a significant amount of work on the application of AIS algorithms, the same cannot be said for theoretical research in the area. Theoretical literature on the topic has only begun to surface in the past few years, with the community now acknowledging that theoretical development is of crucial importance.

This paper attempts both to review works relating to the theoretical aspects of AIS, and to bring together in a single place techniques that have been used to develop theoretical models of immune-inspired algorithms and proofs of convergence where appropriate. We have focussed our discussions on clonal selection based AIS and negative selection based AIS. This is primarily because theoretical studies have thus far concentrated on these aspects of AIS. Despite notable empirical analysis of immune networks [3–6], very little has been done on theoretical aspects of these.

1.1. Why study the immune system for computation?

The immune system is considered to provide both defence and maintenance of the body. There are many reasons why the immune system has been seen as a source of inspiration for the design of novel algorithms and systems. It has many properties which it would be desirable for an artificial system to have, such as the following: self-organisation; learning and memory; adaptation; recognition; robustness and scalability.

In the next section we outline the main components and processes in the immune system and illustrate the above properties.

2. Biological underpinnings of artificial immune systems

The main developments within AIS have focussed on three main immunological theories: clonal selection, immune networks and negative selection. Researchers in AIS have concentrated, for the most part, on the *learning* and *memory* mechanisms of the immune system inherent in clonal selection and immune networks, and the negative selection principle for the generation of detectors that are capable of classifying changes in *self*. In this section, we summarise the immunological ideas that have typically been exploited by the AIS community.

The vertebrate immune system is composed of diverse sets of cells and molecules that work together with other systems, such as the neural and endocrine systems, in order to maintain a homeostatic state within the host. The traditionally held view of the role of the immune system is to protect our bodies from infectious agents such as viruses, bacteria, fungi and other parasites.[1] Such invading agents are known as *pathogens*, and an immune response to a pathogen is provoked by the recognition of an associated molecule called an *antigen*. There are two arms of the immune system: innate and adaptive. Innate immunity is not directed towards specific invaders, but against general pathogens that enter the body [8]. The innate immune system that all vertebrates are born with plays a vital role in the initiation and regulation of immune responses (including the adaptive immune response). Specialised cells of the innate immune system have evolved so as to recognise and bind to common molecular patterns found only in microorganisms. However, the innate immune system by no means provides complete protection of the body, as it is primarily static in nature.

Adaptive or acquired immunity allows the immune system to launch an attack against any invader that the innate system cannot remove [9]. The adaptive system is directed against specific invaders, some of which will have been seen before and others that will not have been previously encountered, and is modified by exposure to such invaders. The adaptive immune system mainly consists of lymphocytes, which are white blood cells, more specifically B and T cells. These cells aid in the process of recognising and destroying specific substances. Adaptive immune responses are normally directed against the antigen that provoked them, and are said to be antigen-specific. The majority of immunological research has focussed on the adaptive immune response. However, there has been a recent focus on the importance of the innate immune system [10].

2.1. Clonal selection theory

The clonal selection theory (CST) [11] is the theory used to explain the basic response of the adaptive immune system to an antigenic stimulus. It establishes the idea that only those cells capable of recognising an antigen will

[1] However, not all immunologists would agree that this is its primary role [7].

proliferate, while those that do not recognise an antigen are selected against. Clonal selection operates on both T cells and B cells. In the case of B cells, when their antigen receptors (antibodies) bind with an antigen, B cells become activated and differentiate into *plasma* or *memory* cells. Prior to this process, clones of B cells are produced, which themselves undergo somatic hypermutation, thus introducing diversity into the B cell population. Plasma cells produce large numbers of antigen-specific antibodies which, in a successful immune response, lead to the removal of the antigen. Memory cells are generally considered to remain within the host and promote a rapid secondary response upon a subsequent encounter with the same (or similar) antigen. This is the phenomenon of acquired immunity.

2.2. Negative selection

As well as responding to antigen coming from external invaders, lymphocytes can react to material coming from the host's own cells. If this leads to a full immune response, this can result in damage to the host organism (this is called auto-immunity). Negative selection is a mechanism employed to help protect the body against self-reactive lymphocytes. Such lymphocytes can occur because the building blocks of antibodies (produced by B cells) are different gene segments that are randomly composed and undergo a further somatic hypermutation process. This process can therefore produce lymphocytes which are able to recognise self-antigens.

The negative selection of T cells occurs within the thymus. The thymus forms a highly impermeable barrier to macromolecules called the blood-thymic barrier. The blood-thymic barrier allows thymocytes (immature T cells) to mature and undergo selection in an environment protected from contact with foreign antigens. During the selection process, antigen presenting cells (APCs) present a self-peptide/major histocompatability complex (MHC) to the T cells. Those that react strongly (bind with high affinity) with the self-peptide/MHC complexes are eliminated through a controlled cell death (called apoptosis). As a result, only those T cells remain which can recognise foreign antigens and are not self-reactive. This process has formed the foundation for a large amount of AIS work, with the seminal paper in the area being that by Forrest et al. [12], and additional significant work has been done recently by Esponda [13].

2.3. Immune networks

In a landmark paper [14], Jerne proposed that the immune system is capable of achieving immunological memory via the existence of a mutually reinforcing network of B cells. This network of B cells occurs due to the ability of paratopes (molecular portions of an antibody) located on B cells, to match against idiotopes (other molecular portions of an antibody) on other B cells. The binding between idiotopes and paratopes has the effect of stimulating the B cells. This is because the paratopes on B cells react to the idiotopes on similar B cells, as they would to an antigen. However, to counter the reaction there is a certain amount of suppression between B cells which acts as a regulatory mechanism. This interaction of B cells within a network was said to contribute to a *stable* memory structure, and account for the retainment of memory cells, even in the absence of antigen. This theory was refined and formalised in some successive works [15,16]. However, the idiotypic network theory is no longer widely accepted by immunologists [17].

3. A brief history of artificial immune systems

The origins of AIS lie in the early theoretical immunology work in [15,16] and [18]. These works investigated the idiotypic network theory (outlined in the previous section). The first paper to propose a link between immunology and computing was [15] which described the parallels between a network of immune cells and the Holland classifier system [19]. One of the earliest applications of an immune inspired approach was in [20] where a simple immune network algorithm was applied to the cart-pole balancing problem. This concept was developed in [21] where further thought was given to learning and adaptation in immune inspired artificial systems. Probably the most influential paper in immune inspired computing was [22] where a simple self/non-self discrimination algorithm was proposed in the context of computer security. This has been extended further in works by Forrest's research group [23–25] which have formed the basis of a great deal of further research by the community on the application of immune inspired techniques to computer security.

Other early work in the area of AIS was undertaken by Ishida [26]. In this work, a distributed diagnosis systems was developed based on the interactions within immune networks. This work formed a basis for subsequent work

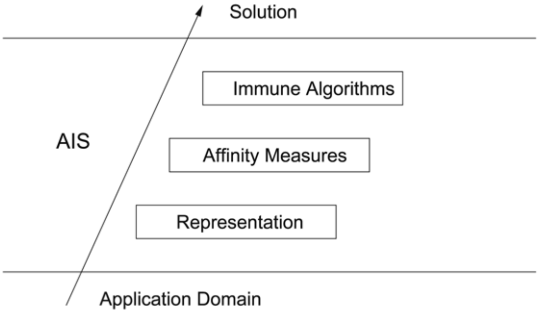

Fig. 1. AIS Layered Framework adapted from [1].

by the same author (e.g. [27,28]) which culminated in a book on immunity-based systems [29], focussing on the maintenance abilities of the immune system and its application in the development of immune inspired systems.

At about the same time as Forrest was undertaking her work, two researchers in the UK, namely Hunt and Cooke, started to investigate the nature of learning in the immune system and how that might be used to create machine learning algorithms [30]. Initial results were very encouraging, and they built on their success by applying the immune ideas to the classification of DNA sequences as either promoter or non-promoter classes [31] and the detection of potentially fraudulent mortgage applications [32].

The work of Hunt and Cooke spawned more work in the area of immune network based machine learning over the next few years, notably in [33] where Hunt and Cooke's system was totally rewritten, simplified and applied to unsupervised learning. Concurrently, similar work was carried out by the authors of [34,35], who developed algorithms for use in function optimisation and data clustering. The work of Timmis on machine learning spawned yet more work in the unsupervised learning domain, in particular dynamic clustering. This has met with some success in works such as [36,37]. At the same time work by Hart and Ross [38] used immune inspired associative memory ideas to track moving targets in databases. In the supervised learning domain, very little was achieved until work in [39] (later augmented in [40]), which developed an immune based classifier known as AIRS. The AIRS system evolves a population of detectors capable of identifying multiple classes, and it has been studied quite extensively [41–43]. The system developed by Watkins was then adapted into a parallel and distributed learning system in [44], for increased scalability.

In addition to the work on machine learning, there has been increased activity in AIS over the past ten years. Recent applications of AIS have included the following areas: computer security, numerical function optimisation, combinatorial optimisation (e.g. scheduling), learning, bio-informatics, image processing, robotics (e.g. control and navigation), adaptive control systems, virus detection and web mining [2]. Rather than outlining all the developments within AIS the reader is directed to a number of review papers and books [45,46,1,47]. Due to a growing amount of work conducted on AIS, the International Conference on Artificial Immune Systems (ICARIS) conference series was started in 2002[2] and has operated in subsequent years [48–53]. This is the best source of reference material to read in order to grasp the variety of application areas of AIS, and also the developments in both algorithms and the theoretical side of AIS.

4. Basics of artificial immune systems

In [1] it was proposed that a framework to design a biologically inspired algorithm, and for engineering an AIS in particular, should require, 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 means.
- Procedures of adaptation that govern the dynamics of the system, i.e., how its behaviour varies over time.

This simple framework results in a layered approach to the development of an AIS (Fig. 1). The authors of [1] argued that in order to build an AIS, an application is a prerequisite. The way in which the components of the

[2] http://www.artificial-immune-systems.org.

system will be represented should be considered on this basis, and depends on the application: so for example, the representation adopted for network traffic may well be different from the representation of a function to optimise. In AIS, the basic components are represented in a *shape space* (see Section 4.1). There are many kinds of shape space, such as Hamming, real-valued and so on, each of which carries its own bias and should be selected with care [54].

Once the representation has been chosen, one or more affinity measures are used to quantify the interactions between the elements of the system. There are many possible affinity measures (which are partially dependent upon the representation adopted), such as Hamming and Euclidean distance metrics. Again, each of these has its own bias, and the affinity function must be selected with great care, as it can affect the overall performance of the system [54].

4.1. Shape space

The notion of shape space was introduced by Perelson and Oster [55] as a quantitative way to describe the affinity between cells and antigens. The basic idea behind shape space is that the degree of matching between receptors on the surface of an immune cell and the *epitopes* (the molecular regions of antigens that are presented to these cell receptors) determines the strength of their binding with one another (affinity), and this is determined by the extent to which the molecules physically fit together, which depends on their shape. The amount of affinity in turn influences which types of cells will have their populations stimulated to expand via clonal selection, and which ones will be suppressed.

A key fitting into a lock provides a very simple analogy for the binding of an antigen to a cell receptor. However, this analogy is too crude, because most doors will only open if the key has exactly the right shape, whereas an epitope can have a less than perfect fit to a receptor and still stimulate the corresponding immune cell, where the amount of stimulation should depend on how well the shapes match. To model the degree of matching, the receptor can be represented by a point (or more generally, a region) in a *shape space* of some dimension, while an epitope is represented by a point in the same space, and the affinity between them is measured by specifying a measure of distance (metric) between the points. Note that in the biological setting the dimension of shape space is not necessarily the same as the dimension of the physical space in which the matching occurs, but rather it is the number of different coordinates that are required to determine the generalized shape of the objects involved. (These coordinates can measure features of the binding sites such as their width or height above the surface, along with other parameters like electric charge.)

As a simple example, we consider a one-dimensional shape space. Suppose that each cell receptor contains a groove of length ℓ, and epitopes that bind to it have a bump of length less than or equal to ℓ which fits into the groove, so that the affinity is just proportional to the length of the bump. Then we have a real shape space where the epitopes are represented by real numbers $x \in [0, \ell]$. The affinity measure (distance) on shape space is just given by the (non-negative) value x, so that every epitope has a maximum affinity ℓ, which puts a threshold on the amount of stimulation that the cells with this receptor can receive.

More generally, the receptors could have various other features or different binding sites for epitopes, so one could use n real numbers to specify the degree of each type of binding, and the shape space would be some subset of $\mathbb{R}^n$ equipped with a suitable measure of distance. For example, for a receptor with coordinates $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and an epitope with coordinates $\mathbf{y} = (y_1, y_2, \ldots, y_n)$, the standard Euclidean metric would give the affinity between them as

$$D(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{j=1}^{n} (x_j - y_j)^2}.$$

In this way one can assign an explicit affinity to every pair for shapes in the space [56], but there are many other possible choices of metric. In fact, a simpler approach (which was originally adopted by Perelson and Oster) is to construct a “recognition ball” $B_\epsilon(\mathbf{x})$ of radius ϵ around the receptor at point $\mathbf{x}$, that is

$$B_\epsilon(\mathbf{x}) = \{\mathbf{y} \mid D(\mathbf{x}, \mathbf{y}) < \epsilon\},$$

and assume that receptors have a strong enough affinity to generate an immune response with all epitopes lying inside their respective recognition balls, while epitopes outside each ball generate no response with the corresponding receptor. In that case one can take the simplified affinity measure

$$\tilde{D}(\mathbf{x}, \mathbf{y}) = \begin{cases} 1 & : \quad \mathbf{y} \in B_\epsilon(\mathbf{x}) \\ 0 & : \quad \mathbf{y} \notin B_\epsilon(\mathbf{x}) \end{cases}$$

$$\begin{array}{cc}
\Sigma = \{0,1\} & \Sigma = \{A,C,G,T\} \\
 & \\
000\ldots000 & AAA\ldots AAA \\
000\ldots001 & AAA\ldots AAC \\
\ldots\ldots\ldots\ldots & \ldots\ldots\ldots\ldots \\
\ldots\ldots\ldots\ldots & \ldots\ldots\ldots\ldots \\
\underbrace{\underline{111\ldots111}}_{L} & \underbrace{\underline{TTT\ldots TTT}}_{L}
\end{array}$$

Fig. 2. Different Hamming shape spaces.

for each receptor at point **x** in shape space. In general the size of the recognition ball (the radius ϵ) can depend on the receptor. Moreover, the recognition region does not need to be exactly spherical, but its volume relative to that of the entire space should provide a measure of the probability of recognition (see Section II in [57] for more details).

In the context of computational models of real immune systems, as well as in artificial immune system algorithms, it is more appropriate to consider a discrete shape space rather than a continuous one. Clearly, if one represents a continuous shape space on a computer, then some sort of discrete approximation will result, but if this is coded with a floating point representation of the real numbers then some undesirable features can arise: the floating point representation amplifies small-scale features but gives rise to a coarse-grained view at large scales. Despite these drawbacks, many AIS algorithms have been coded in terms of real shape spaces represented in floating point e.g. [4]. However, for computational purposes it seems more sensible to start from a discrete model of shape space in the first place. One common example of this is the Hamming shape space, which we now describe.

In AIS algorithms, all artificial immune elements, like antibodies and antigens, are represented by points in a shape space. A detailed overview is provided in [1]. The Hamming shape space Σ^L is built out of all elements of length L over a finite alphabet Σ.

In Fig. 2, two Hamming shape spaces for different alphabets and alphabet sizes are presented. On the left hand side, a Hamming shape space of dimension L defined over the binary alphabet is shown. On the right hand side, a Hamming shape space defined over an alphabet with four symbols (the DNA bases Adenine, Cytosine, Guanine, Thymine) is presented. A formal description of antigen-antibody interactions not only requires an encoding of points, but also appropriate affinity functions. Percus et. al [58] proposed the *r-contiguous* matching rule for abstracting the affinity of an antibody needed to recognise an antigen.

Definition 1. An element $\mathbf{e} \in \Sigma^L$ with $\mathbf{e} = (e_1, e_2, \ldots, e_L)$ and detector $\mathbf{d} \in \Sigma^L$ with $\mathbf{d} = (d_1, d_2, \ldots, d_L)$, match according to the r-contiguous rule, if a position p exists where $e_i = d_i$ for $i = p, \ldots, p + r - 1$, $p \leq L - r + 1$.

Informally, two elements of the same length match if at least r contiguous characters are identical.

4.2. Clonal selection algorithms

There are many clonal selection based algorithms in the literature, most of which have been applied to optimisation problems (e.g. CLONALG [34] and opt-IA is used in [59], and the B Cell Algorithm in [60]), and to multi-objective optimisation (see [61,62]). From a computational perspective, the clonal selection idea leads to algorithms that iteratively improve candidate solutions to a given problem through a process of cloning, mutation and selection (in a manner similar to genetic algorithms [63]). In this section we describe a clonal selection-based algorithm that has been used as a basis for theoretical studies [64].

4.2.1. Simple clonal selection algorithm

The B Cell Algorithm (BCA) is an optimisation algorithm that was introduced in [60]; an outline of it is shown in Algorithm 1. Simply stated, through a process of evaluation, cloning, mutation and selection, the algorithm evolves a population of individuals (B cells) towards a global optimum. Each member of this population can be considered as an independent entity, i.e. there are no interactions between members of the population. Each B cell within the population is represented by an L-dimensional vector **v** in a binary Hamming shape space. (In the original implementation [60], strings of 64 bits were used.) The objective function g is evaluated on each B cell **v** within the population P to produce the value $g(\mathbf{v})$.

After evaluation by the objective function, a B cell $\mathbf{v}$ is cloned to produce a clonal pool, C. It should be noted that there exists a clonal pool C for each B cell within the population, and also that all the adaptation (mutation) takes place within C. The size of C is typically the same size as the population P (but this does not have to be the case).

In order to maintain diversity within the search, one clone is selected at random and each element in the vector is randomized. Each B cell $\mathbf{v}' \in C$ is then subjected to a contiguous region somatic hypermutation operator (CRHO). This mutation mechanism selects a contiguous region within the bitstring that is subject to mutation at a uniform rate ρ (i.e. a number between 0 and 1 which is the probability for each bit in the contiguous region to flip). Due to the bias in the way that the region is formed, values of ρ used in more conventional operators are much lower than rates found to work well in the CRHO. Details of the CRHO itself and findings about appropriate values of mutation rate ρ were first explored empirically in [60], and initial theoretical work was presented in [64]. It should also be noted that unlike the case of genetic algorithms, no cross-over is employed between candidate solutions or their respective clone populations. For selecting the members from C an elitist mechanism is employed. The BCA typically uses a distance function as its stopping criterion. In the case of benchmark problems, the algorithm can be stopped when the best solution is within a certain prescribed distance from the known optimum. For a general problem, one can stop the algorithm if the value of the objective function has not improved after a set number of iterations.

```
Algorithm 1: B Cell Algorithm
  input  : $g(\mathbf{v})$ = function to be optimised
  output: $P$ = set of solutions for function
  begin
      1. Create an initial population $P$ of individuals in shape–space $\Sigma^L$
      2. For each $\mathbf{v} \in P$, evaluate $g(\mathbf{v})$ and create clone population $C$
      3. Select a random member of $\mathbf{v}' \in C$ and apply the contiguous region hypermutation operator
      4. Evaluate $g(\mathbf{v}')$; if $g(\mathbf{v}') > g(\mathbf{v})$ then replace $\mathbf{v}$ by clone $\mathbf{v}'$
      5. Repeat steps 2-4 until stopping criterion is met
  end
```

4.3. Immune network algorithms

The algorithm aiNet, inspired by Jerne's postulated immune network theory, was developed by de Castro and von Zuben [65,66] with the aim of finding a reduced set of points that closely represents the set of input points, or in other words, finding a compressed input set representation with less redundancy. The process within aiNet that evolves a population towards a set of effective detectors is very similar in nature to a clonal selection approach: the main difference being that there are interactions between members of the population via a supression mechanism that removes members whose match between themselves and a training data item falls below a certain threshold. A simplified version of aiNet can be described as follows:

```
Algorithm 2: Immune Network Algorithm
  input  : $G$ = pattern to be recognised, $N$ a set of random detectors, $n$ number of best antibodies
  output: $M$ = set of generated detectors capable of recognising input pattern
  begin
      1. Create an initial random population $B$
      2. For each pattern to learn
      2.1 Determine inverse distance for pattern in $B$ to each member of $N$
      2.2 Select $n$ members of $B$ with the best match to each pattern
      2.3 Clone and mutate each $n$ in proportion to how good the match to the pattern is
      2.4 Retain the highest matching of $n$ and place in a set $M$
      2.5 Perform network dynamics in $M$ to remove weak members of $M$
      2.6 Generate $b$ random elements and place in $B$
      3 repeat
  end
```

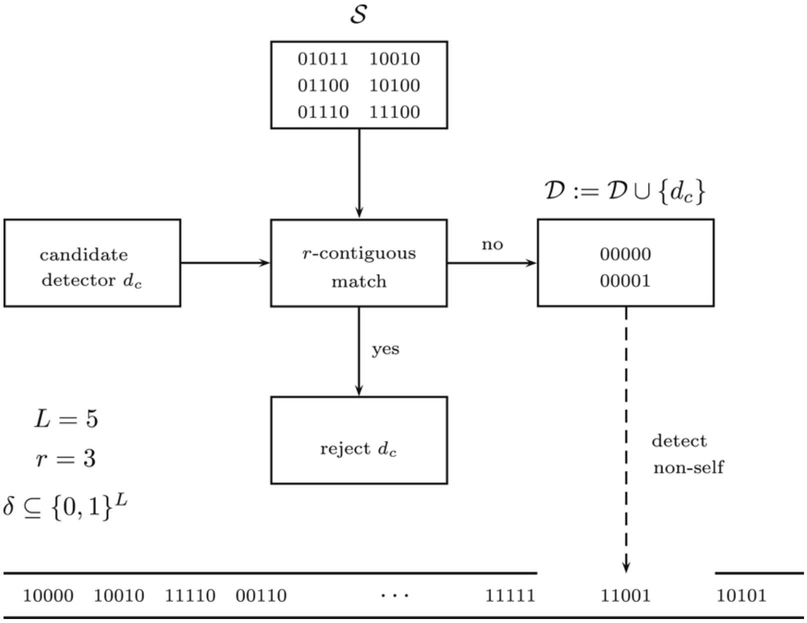

Fig. 3. Principle of negative selection.

4.4. *Negative selection algorithm*

The basic idea of a negative selection algorithm (Fig. 3) is to generate a number of detectors in the complementary set N, and then apply these detectors to classify new data as self or non-self [12]. Negative selection algorithms have been very extensively used in AIS research and have undergone a number of augmentations throughout the years. Here we describe the negative selection algorithm originally proposed by [12] which underpins all of the subsequent negative selection work. This will be further analysed in Section 5.4. We can summarise the algorithm in the following steps:

Given a shape–space Σ^L and a self set $S \subset \Sigma^L$ we define the non-self set $N \subset \Sigma^L$ to be the complement $N = \Sigma^L \setminus S$, so that

$$\Sigma^L = S \cup N \quad \textit{and} \quad S \cap N = \emptyset.$$

Algorithm 3: Negative Selection Algorithm

```
input  : $S$ = set of self elements
output: $D$ = set of generated detectors
begin
    1. Define self as a set $S$ of elements in shape–space $\Sigma^L$.
    2. Generate a set $D$ of detectors, such that each fails to match any element in $S$.
    3. Monitor data $\delta \subseteq \Sigma^L$ by continually matching the detectors in $D$ against $\delta$. If any detector match with $\delta$,
    classify $\delta$ as a non-self, else as self.
end
```

5. Analysis of artificial immune algorithms

Compared with the large number of applications of AIS, very little theoretical work has been carried out. Above we have given an example of each of the three main types of AIS algorithm, namely clonal selection algorithms, immune network algorithms, and negative selection algorithms. In this section, we review the existing theoretical analysis that has been performed for these three different types of algorithm, and point out where such analysis is lacking. We describe the available results of theoretical analysis of the foregoing examples in some detail, and then merely indicate how this can be extended or generalized to other algorithms of the same type. Often the necessary generalization has still to be performed, so we also discuss some of the main open problems in each area.

Within the field of AIS there has been extensive use of clonal selection-based algorithms, which are fundamentally stochastic in nature. Therefore, in order to model AIS algorithms (and clonal selection ones in particular) it makes sense to consider the evolution of a population belonging to a discrete state space and changing according to probabilistic rules. As long as the probabilities for transitions to a new state depend only on the current state of the system (and not on the previous history), all the properties of a Markov chain are satisfied, so it is natural to a describe AIS clonal selection algorithms in these terms. Almost all immune network algorithms also include some form of clonal selection in order to update the population, so the techniques of Markov chain theory are relevant in that context as well. Therefore, in the next subsection we introduce some basic concepts and terminology for Markov chains, which are required for what follows.

5.1. *Markov chains for clonal selection*

For a general clonal selection algorithm, one has a population of N_p individuals, each of which is an element $\mathbf{x} \in \Sigma^L$, where Σ is some finite alphabet (and usually we take $\Sigma = \{0, 1\}$). Hence the state of the system at any time is specified by a point $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_{N_p})$ in the state space Σ^{LN_p}. This state space is finite, and so we can assign a number to each state $\mathbf{X}$ by counting off the elements of Σ^{LN_p}. Thus we identify each state with a number from 1 to $\mathcal{M}$, where $\mathcal{M} = |\Sigma|^{LN_p}$ is the number of possible states. Since the algorithm is stochastic, at each time step (or generation) t there is a random variable X_t, taking values from 1 to $\mathcal{M}$, which corresponds to the state of the system. The complete behaviour of the algorithm is encoded in the infinite sequence of random variables $X_0, X_1, X_2, \ldots$, and ideally we would like to understand the probability distribution of X_t and how it behaves as $t \to \infty$.

Let $\mathbf{v}_t$ denote the $\mathcal{M}$-dimensional row vector for the probability distribution of X_t, so that the jth component of this vector is $v_{t,j} = P(X_t = j)$ (i.e. the probability that the system is in state j at time t). For a generic clonal selection algorithm, the probability that the system moves to state k at time $t + 1$, given that it is in state j at time t, does not depend on t. From the rules of conditional probability, the probability distribution at time $t + 1$ is given in terms of that at the preceding time t as follows:

$$P(X_{t+1} = k) = \sum_{j=1}^{\mathcal{M}} P(X_{t+1} = k | X_t = j)\, P(X_t = j);$$

or more concisely this is

$$v_{t+1,k} = \sum_{j=1}^{\mathcal{M}} v_{t,j}\, P_{jk}, \tag{5.1}$$

where $P_{jk} = P(X_{t+1} = k | X_t = j)$ is the probability of transition from state j to k, and since this is independent of t we have also

$$P_{jk} = P(X_1 = k | X_0 = j).$$

The above properties mean that the sequence of variables X_t form a Markov chain, and because of the t-independence of P_{jk} it is called an homogeneous chain.

The Eq. (5.1) can be summarized even more neatly in matrix form, as

$$\mathbf{v}_{t+1} = \mathbf{v}_t\, \mathbf{P}, \tag{5.2}$$

where $\mathbf{P} = (P_{jk})$ is the transition matrix, that is the $\mathcal{M} \times \mathcal{M}$ matrix of transition probabilities. It follows immediately from (5.2) that

$$\mathbf{v}_t = \mathbf{v}_0\, \mathbf{P}^t, \tag{5.3}$$

so that the asymptotic distribution of the chain depends on the limiting behaviour of $\mathbf{P}^t$ as $t \to \infty$. A single state j of the chain is said to be *absorbing* if $P_{jk} = 0$ for all $k \neq j$: in other words, once the system reaches that state, it will never leave. A Markov chain is said to be absorbing if it has at least one absorbing state, and from every state it is possible to reach an absorbing state.

In order to understand the convergence proof in the next subsection, it is also helpful to introduce one more definition. A state j of the chain is called *persistent* if

$$P(X_t = j \text{ for some } t \geq 1 | X_0 = j) = 1, \tag{5.4}$$

and otherwise it is called *transient*. More intuitively, if the system starts off in a persistent state, then it must return there (with probability 1), while if the state is transient then the system need not return to it. For more details on the theory of Markov chains, the reader is referred to chapter 6 of [67].

5.2. Analysis of clonal selection: The B cell algorithm

Clonal selection algorithms are usually applied to function optimisation and search problems. In particular, the B Cell algorithm, as described in Section 4.2.1, provides one example of an immune algorithm used for optimisation, while the more general problem of multiobjective optimisation has been tackled with MISA [62], a multiobjective artificial immune system algorithm. For problems of this kind, one of the first questions of interest is whether the algorithm is convergent, in the sense that it finds (at least one) global optimum solution with probability one, in the limit as $t \to \infty$. The first paper to address this problem was [68], which used Markov chain theory to prove convergence of MISA, with the proviso that an elitist memory set must be maintained. In [64], after modelling the associated hypermutation operator, the same method of proof was adapted and simplified in order to prove convergence for the case of the B cell algorithm, which we describe shortly.

Before proceeding to give specific details for the B cell algorithm, it is worth remarking on the similarities between clonal selection-based AIS algorithms and genetic algorithms (GAs). The paper [64] adopts a novel method for the construction of the transition matrix which allows a more transparent analysis of how the features of the Markov chain relate to the operators in the algorithm. Evolutionary algorithms have been extensively modelled with Markov chains for some time. For instance, Nix and Vose presented an exact Markov chain model of a simple genetic algorithm [69], which was then analyzed in depth in several papers by Vose; these results are summarized in [70]. The model of Nix and Vose does not show the simple GA to be convergent, although it is clear from other results (such as those in [68]) that it could be shown to be convergent with the additional condition that an elitist memory set should be maintained. In fact, more recent results on applying Markov chains to convergence of GAs, as presented by Rudolph [71], give quite general sufficient conditions for their convergence. This means that in many cases one can merely check that these conditions are satisfied, and there is no need to do any detailed modelling unless the conditions do not hold. In that sense, there is already a general framework for understanding the convergence of GAs.

In a similar vein, the paper [68] calls for a general mathematical framework that would allow an entire family of artificial immune algorithms to be shown to be convergent instead of a specific proof for each algorithm. It seems that such a framework has just appeared. Very recently, the authors heard the results of [72], which adapt the criteria of Rudolph to the setting of a generic clonal selection-type algorithm, called the Immune Algorithm (IA). This IA includes the possibility of a variety of different schemes for hypermutation, aging and so on, and two sufficient conditions for convergence are given in [72]. In particular, it appears that the convergence of the B Cell Algorithm can also be proved by minor modifications of the approach in that paper. However, here we shall just review the approach originally taken in [64], before suggesting some possible extensions of this work.

One of the most interesting features of the B Cell Algorithm (BCA) is the contiguous region hypermutation operator, for which a formal model was presented in [64], as a prelude to the proof of convergence. From the structure of the mutation operator, one can understand the salient features of the transition matrix $\mathbf{P}$ associated with the algorithm. As previously mentioned in Section 4.2.1, the BCA maintains two populations of candidate solutions: the memory set and the clonal pool. In the case where the clonal pool C has only one member, it is possible to obtain formulae for the probabilities of all possible mutation masks. (Although it is more complicated, this can be extended to the case of a clonal pool of arbitrary size.) These probabilities are deduced by counting all possible ways each mutation mask can occur, for an exhaustive set of mutation masks. Letting f_T be the probability of transition from zero to some number T (the Tth state of a string of length L), we have

$$f_T = \frac{1}{L^2}\left[\sum_{n=1}^{a}\sum_{m=b}^{L-1}(1-\rho)^{m+1-n-k}\rho^k + \sum_{n=1}^{a} n(1-\rho)^{L+1-n-k}\rho^k\right], \tag{5.5}$$

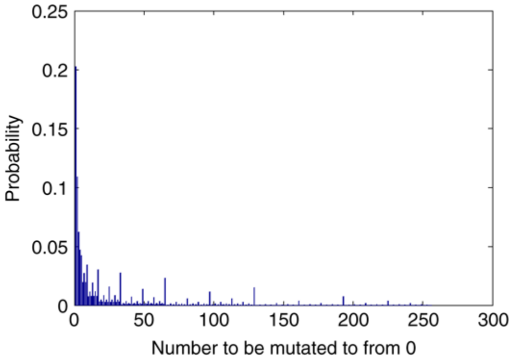

Fig. 4. Theoretically determined probabilities of generating mutation masks from the CRHO: String Length $L = 8$, Probability of mutation $\rho = 0.5$. Taken from [64].

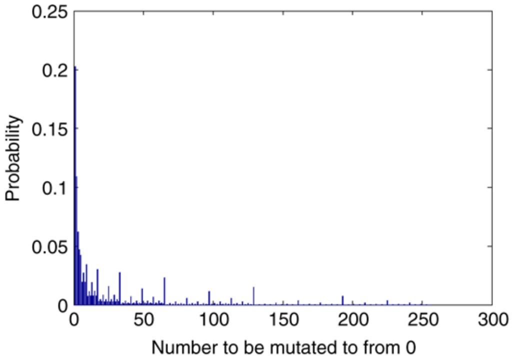

Fig. 5. Probabilities determined experimentally by averaging one million results from the implemented CRHO: String Length $L = 8$, Probability of mutation $\rho = 0.5$. Taken from [64].

where a is the bit position of the first flipped bit, b is the bit position of the last flipped bit (both measured starting from the most significant bit), k is the total number of bits that must be flipped to mutate from 0 to T, and ρ is the probability of a bit being mutated given it is in the contiguous region. In this setting we have $0 \leq T \leq 2^L - 1$, $a \leq b \leq L$ and $0 \leq \rho \leq 1$.

Eq. (5.5) can be used to generate the probabilities of all non-zero mutation masks, and this formula is general (for a clonal pool of one), and can also be applied to the Gray code or any other binary system. The probability of a mutation mask is given as a polynomial in ρ. The contiguous region hypermutation operator was also implemented for some particular numerical values, and the theoretical results were found to match almost perfectly with experiment (see Figs. 4 and 5 respectively, reproduced from [64]).

The proof of convergence in [64] is based on the fact that the BCA corresponds to an absorbing Markov chain (see Section 5.1) with a non-zero one step transition probability from all points in the space to any of the global optima. This is due to the elitist mechanism in the BCA. This mechanism essentially ensures that the Markov chain is reducible, as opposed to an irreducible chain as is often see with genetic algorithms [69]. In the terminology of Markov chains, all non-optimal states of the BCA are transient. Moreover, all the optima are absorbing states.

Proposition 1. *All non-optimal states are transient for* $0 < \rho < 1$.

Proof. Under the condition $0 < \rho < 1$, the sample matrix contains only non-zero elements. By inspection of Eq. (5.5) it is clear that if $f_T > 0$ if $(1 - \rho) > 0$ and $\rho > 0$. Thus, we impose the condition $0 < \rho < 1$. (The formulae in Eq. (5.5) need to be modified in the case when the clonal pool C has more than one member but the condition for ρ is the same.) Hence it is possible to reach the absorbing state in one step from any initial state, and this is irrespective of the details of the objective function. Hence, for a non-optimal state j the probability of remaining in that state for one time step is $p = P_{jj} < 1$. Once a state has been left for a state of higher affinity, it can never return to a previously occupied state, due to the possible transit matrix forbidding transitions to states of equal or lower affinity. Hence it follows that if $P_{jk}(n)$ is the j, k entry of $\mathbf{P}^n$, then

$$\sum_{n=0}^{\infty} P_{jj}(n) = \sum_{n=0}^{\infty} p^n = \frac{1}{1-p} < \infty \tag{5.6}$$

hence the state j is transient (see p. 122 in [67], for instance). □

The general theory of Markov chains is particularly effective in the case of irreducible chains. The chain corresponding to the BCA is reducible, however: by the Decomposition Theorem [67] it can be partitioned into the set of transient non-optimal states together with the (disjoint) closed sets of absorbing states corresponding to the optima. Adopting the terminology of [73], we can say that the non-optimal states of the BCA are inessential, in the sense that for all such non-optimal states j there exists another state k such that j can make a transition to k but not vice-versa: it is sufficient to choose any state k with a larger value of the objective function. Then if there are K optima we can partition the transition matrix $\mathbf{P}$ as follows:

$$\mathbf{P} = \left(\begin{array}{c|c} \mathbf{1} & \mathbf{0} \\ \hline \mathbf{R} & \mathbf{Q} \end{array}\right), \tag{5.7}$$

where $\mathbf{1}$ denotes the $K \times K$ identity matrix, $\mathbf{Q}$ corresponds to the transitions between the inessential states and $\mathbf{R}$ corresponds to transitions from inessential to optimal states. It follows from standard properties of stochastic matrices that the powers $\mathbf{Q}^t \to 0$ as $t \to \infty$. Using this partitioning, we find the following result.

Proposition 2. *All optima are absorbing states.*

Proof. By definition, the possible transit matrix prohibits transition from a global optimum to any other state, even another global optimum. Therefore once the algorithm enters a global optimum it satisfies the condition for persistence. Clearly for an optimum state j we have $p = P_{jj} = 1$ (i.e. once an optimum is reached then one remains there with probability one), so the corresponding sum as in Eq. (5.6) diverges and the optima are all persistent states. They are also absorbing, since they do not communicate with any other state. □

Theorem 1. *The BCA converges to the optima of the objective function, for* $0 < \rho < 1$*, where* ρ *(the mutation rate) is the probability of mutation for bits contained within the contiguous region.*

Proof. We have the state vector calculated from powers of the transition matrix according to (5.3). Now

$$\mathbf{P}^t = \left(\begin{array}{c|c} \mathbf{1} & \mathbf{0} \\ \hline \mathbf{R}_t & \mathbf{Q}^t \end{array}\right) \tag{5.8}$$

for some matrix $\mathbf{R}_t$ constructed from sums of powers of $\mathbf{Q}$ acting on $\mathbf{R}$. It follows from standard properties of stochastic matrices that the powers $\mathbf{Q}^t \to 0$ as $t \to \infty$, the powers of $\mathbf{P}$ take the form

$$\mathbf{P}^\infty = \left(\begin{array}{c|c} \mathbf{1} & \mathbf{0} \\ \hline \mathbf{R}_\infty & \mathbf{0} \end{array}\right) \tag{5.9}$$

Acting with this matrix on the initial state vector $\mathbf{v}_0$ we see that the probability of being in any of the non-optimal transient states tends to zero as $t \to \infty$, and hence the probability of ending up in an optimum tends to one, as required. □

We have given a fair amount of details about the structure of the Markov chain associated with the BCA. According to the recent results in [72], some of these details can be bypassed if one only wishes to know whether an algorithm of this kind converges or not. However, convergence only describes the limiting distribution as $t \to \infty$. A more useful sort of result, for practical applications of clonal selection algorithms, would be to know an upper bound for the number of iterations, t_{max} say, for which the algorithm converges with some certainty (say, 90 per cent confidence).

The authors of [72] give some preliminary estimates of such bounds for the generic IA. Yet as they point out, without any more specific information on the details of the algorithm or the type of objective function, such estimates are too crude to be of practical use. Thus a major theoretical challenge for the future is to give sharper bounds for the performance of an IA (or of other clonal selection mechanisms) when applied to some specific functions, and to see how this depends on the form of the mutation operator and other features of the algorithm. Ideally, one would like to have an idea of which type of IA would be most effective for a particular class of problems.

5.3. Theoretical issues for immune network algorithms

A lot of the original inspiration for AIS algorithms came from earlier models in theoretical immunology (see [15]), which were usually formulated in terms of nonlinear differential equations for networks of cell and antigen populations, or in terms of automata or difference equations. The extensive review [57] presents the state of the art in such models a decade ago. There is a great deal of dynamical systems theory that describes the behaviour of fixed networks, e.g. cell networks modelled by systems of differential equations. However, for AIS network algorithms so far only empirical analysis has been performed [3–6], due to the changing nature of the networks appearing therein. In this subsection we merely highlight some of the challenges posed by immune network algorithms from a theoretical point of view.

The theory of dynamical systems contains a wide variety of tools for analysing nonlinear models (see [74–76] for instance). The natural question as to whether the same tools could be applied to AIS algorithms, which are inspired by nonlinear models of biology, was raised in [77]. However, while the concept of a phase space (or state space) is natural for modelling both biological and artificial immune systems, there are some fundamental differences between the two. The first thing to observe is that the number of immune cells in the human body is absolutely vast (of the order of 10^{12} lymphocytes according to current estimates), which makes it sensible to use continuous models of lymphocyte populations (i.e. differential equations), whereas the number of “cells” making up the clone population in existing AIS algorithms is usually in the tens or hundreds at most. Thus the effects of the discreteness of the population play an important role in AIS. Another essential observation is that most (continuous or discrete) dynamical models in immunology, such as those reviewed by Perelson and Weisbuch [57], are completely deterministic, while within AIS there has been extensive use of clonal selection-based algorithms, which are fundamentally stochastic in nature.

Immune network algorithms have two main parts: a system of differential equations or difference equations, describing the interactions in the network and how they evolve with time; and a set of rules for modifying or updating not only the interaction strengths, but also the number of different populations or agents in the network (cf. algorithm 2 in Section 4.3). The differential or difference equations constitute a dynamical system in the standard sense, in a phase space of finitely many dimensions (or degrees of freedom), and these can be analyzed by standard techniques e.g. stability analysis in phase space, Lyapunov functions, and so on. However, it was already pointed out by the authors of [15] that there is not yet any good theory to describe features like steady states and their stability in the case where the size of the network can change with time. In addition, the changes are usually stochastic, according to clonal selection rules. (However, some network algorithms, such as those considered by Tarakanov [78,79], have a phase space that changes deterministically.) One could also imagine that if the populations change very rapidly, then the network behaves much more like a (continuous) Markov process than a deterministic one, even for short times. Therefore, while it may be possible to get some idea of the behaviour of an immune network over short timescales (where the size of the network remains constant), over long times there is currently no basic theory to predict the behaviour of such a system. This represents a major challenge for dynamical modelling of AIS network algorithms.

5.4. Theoretical issues for the negative selection algorithm

The negative selection algorithm (see Section 4.4) is one of the most frequently applied algorithms for anomaly detection problems [2]. However, for real-world problems it turns out that finding efficiently r-contiguous detectors seems to be computationally infeasible if the matching length r is large (e.g. $r > 32$). In this section, we present an analysis of generating detectors randomly and then the subsequent (and surprising link) to a special type of the k-CNF satisfiability problem. We will show that finding detectors is equivalent of finding assignment sets for k-CNF instances (which is $\mathcal{NP}$-complete for $k > 2$).

5.4.1. *Probability of detection for random detector generation*

To approximate the probability that a randomly drawn detector from shape space $\{0, 1\}^L$ recognizes (when using the r-contiguous matching rule) a randomly drawn antigen, one can apply results from probability theory, namely recurrent events and renewal theory [80]. In Feller's textbook [80] on probability theory an equivalent problem[3] is formulated as follows:

> "*A sequence of n letters S and F contains as many S-runs of length r as there are non-overlapping uninterrupted blocks containing exactly r letters S each.*"

Given a Bernoulli trial with outcomes S (success) and F (failure), the probability of no success running of length r in L trials is according to Feller

$$\frac{1-px}{(r+1-rx)q} \cdot \frac{1}{x^{L+1}} \tag{5.10}$$

where

$$p = q = \frac{1}{2} \quad \text{and} \quad x = 1 + qp^r + (r+1)(qp^r)^2 + \cdots$$

as term (5.10) gives the probability of no success run of length r in L trials, the correct approximation that a random detector recognizes with r-contiguous matching rule a random antigen results in

$$P_{WF} = 1 - \left(\frac{1-px}{(r+1-rx)q} \cdot \frac{1}{x^{L+1}} \right). \tag{5.11}$$

According to results provided in [12], one can calculate the number of r-contiguous detectors which are required to detect non-self bit-strings by allowing with probability P_{fail} that the randomly generated detectors will not match.

Let $\{0, 1\}^L$ be a Hamming shape–space, $\{D_0, D, S\} \subseteq \{0, 1\}^L$ and

$$\begin{aligned}
|D_0| &= \text{number of initial detectors} \\
& \text{(before negative selection)} \\
|D| &= \text{number of detectors} \\
& \text{(after negative selection)} \\
|S| &= \text{number of self elements in } S \\
P_{WF} &= \text{probability according to (5.11)} \\
P_{\neg S} &= \text{probability of a random element from} \\
& \{0,1\}^L \text{ not matching any element from } S \\
&= (1 - P_{WF})^{|S|} \approx \mathrm{e}^{-P_{WF} \cdot |S|} \\
P_{fail} &= \text{probability that } |D| \text{ detectors fail to} \\
& \text{detect a non-self bit-string} \\
&= (1 - P_{WF})^{|D|} \approx \mathrm{e}^{-P_{WF} \cdot |D|}.
\end{aligned}$$

Given a pre-defined number of randomly drawn initial detectors $|D_0|$, $|S|$ and P_{WF}, one obtains the number of suitable detectors $|D|$ not matching any element in S as follows:

$$|D| = |D_0| \cdot P_{\neg S}. \tag{5.12}$$

The number of detectors $|D|$ that fails to detect a non-self bit-string with probability P_{fail} is

$$|D| = -\frac{\ln(P_{fail})}{P_{WF}}. \tag{5.13}$$

[3] The link between recurrent events, renewal theory and the r-contiguous matching rule was discovered originally by Percus et al. [58] and rediscovered by Ranang [81]. Percus et al. presented in [58] an approximation which is only valid for $r \geq L/2$, but mentioned the full approximation for $1 \leq r \leq L$ indirectly by mentioning the name de Moivre and citing Uspensky's textbook (see p. 77 in [82]).

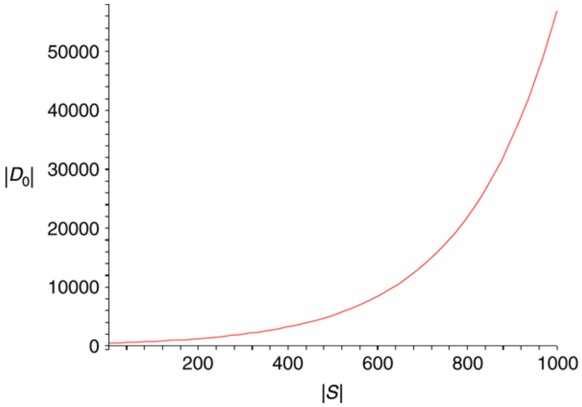

Fig. 6. Coherence between the number of self elements $|S|$ and the number of initial detectors $|D_0|$ according to (5.14) for $L = 49, r = 12$ and $P_{fail} = 0.1$. The values are chosen as proposed in [25].

Combining (5.12) and (5.13)

$$|D_0| = \frac{-\ln(P_{fail})}{P_{WF} \cdot P_{\neg S}} \tag{5.14}$$

one obtains the number of initial detectors $|D_0|$. That means, for detecting a non-self bit-string by allowing with probability P_{fail} that $|D|$ detectors fail to detect a non-self bit-string, one requires a size of $|D_0|$ initial detectors.

By transforming (5.14) into $(e^{P_{WF} \cdot |S|} \cdot (-\ln(P_{fail})/P_{WF}))$ it appears that (5.14) grows exponentially in $|S|$ (see also Fig. 6), i.e. this random search approach becomes infeasible for large $|S|$.

5.4.2. *The Link between r-contiguous Detectors and k-CNF Satisfiability*

We have seen in the previous Section 5.4.1 that the random search for detectors is not a feasible approach for fixed L, r and growing $|S|$. In this section we show how r-contiguous detectors are related to the k-CNF satisfiability problem.

The Boolean satisfiability problem is a decision problem and can be formulated in terms of the language SAT [83]. An instance of SAT is a Boolean formula ϕ composed of $\wedge$ (AND), $\vee$ (OR), $\bar{\cdot}$ (NOT), $\rightarrow$ (implications), $\leftrightarrow$ (if and only if), variables $x_1, x_2, \ldots,$ and parentheses. In SAT problems, one has to decide if there is some assignment of *true* and *false* values to the variables that will make the Boolean formula ϕ true. In the following sections, we will focus on Boolean formulas in conjunctive normal form.

A Boolean formula is in conjunctive normal form (CNF), if it is expressed as an AND-combination of clauses and each clause is expressed as an OR-combination of one or more literals. A literal is an occurrence of a Boolean variable x or its negation $\overline{x}$.

A Boolean formula is in k-CNF, if each clause has exactly k distinct literals. A k-CNF Boolean formula is *satisfiable* if there exists a set of values ($0 \equiv$ false and $1 \equiv$ true) for the literals that causes it to evaluate to 1, i.e. the logical value *true*.

We will now consider a special subset of Boolean formulas in k-CNF which are defined as follows:

Definition 2. A k-CNF Boolean formula ϕ_{rcb} is in L-k-CNF, when ϕ_{rcb} has $(L - k + 1)$ clauses $C_1, C_2, \ldots, C_{L-k+1}$ for $1 \le k \le L$ and $k - 1$ equal literals in C_i, C_{i+1} for $i = 1, 2, \ldots, L - k$

$$\begin{aligned}
C_1 &= (x_1 \vee x_2 \vee \cdots \vee x_k) \\
C_2 &= (x_2 \vee x_3 \vee \cdots \vee x_{k+1}) \\
&\vdots \\
C_{L-k+1} &= (x_{L-k+1} \vee x_{L-k+2} \vee \cdots \vee x_L).
\end{aligned}$$

Recall r-contiguous detectors are bit-strings from $\{0, 1\}^L$ which do not match any bit-strings of length L from S with the r-contiguous matching rule. We subsequently show a transformation of arbitrary bit-strings from S into L-k-CNF Boolean formulas.

Let $b \in \{0, 1\}$ and $\mathcal{L}(b)$ a mapping defined as:

$$\mathcal{L}(b) \rightarrow \begin{cases} x & \text{if } b = 0 \\ \overline{x} & \text{otherwise} \end{cases}$$

where $x, \overline{x}$ are literals.

Let $k, L \in \mathbb{N}$, where $k \leq L$ and $s \in \{0, 1\}^L$, where $s[i]$ denotes the bit at position i of bit-string s, and $\mathcal{C}(s, k)$ a L-k-CNF mapping defined as:

$$\begin{aligned}
&\mathcal{C}(s, k) \rightarrow \\
&(\mathcal{L}(s[1]) \vee \mathcal{L}(s[2]) \vee \cdots \vee \mathcal{L}(s[k])) \wedge \\
&(\mathcal{L}(s[2]) \vee \mathcal{L}(s[3]) \vee \cdots \vee \mathcal{L}(s[k+1])) \wedge \\
&\qquad\qquad\qquad\vdots \\
&(\mathcal{L}(s[L-k+1]) \vee \cdots \vee \mathcal{L}(s[L]))\,.
\end{aligned}$$

For the sake of clarity we denote a Boolean formula in L-k-CNF which is obtained by $\mathcal{C}(s, k)$ for $s \in S$ as ϕ_{rcb}. Moreover we denote a Boolean formula $\bigwedge_{i=1}^{|S|} \phi_{rcb}^i$ which is obtained by $\mathcal{C}(s_1, k) \wedge \mathcal{C}(s_2, k) \wedge \cdots \wedge \mathcal{C}(s_{|S|}, k)$ for $|S| \geq 1$ and all $s_i \in S,\ i = 1, \ldots, |S|$ as $\widehat{\phi}_{rcb}$. If $|S| = 1$, then $\phi_{rcb} \equiv \widehat{\phi}_{rcb}$.

Proposition 3. *Given a shape–space $\{0, 1\}^L$, a set $S \subset \{0, 1\}^L$ and the set $D \subset \{0, 1\}^L$ which contains all generable r-contiguous detectors, which do not match any bit-string from S. The Boolean formula $\widehat{\phi}_{rcb}$ which is obtained by $\mathcal{C}(s, r)$ for all $s \in S$ is satisfiable only with the assignment set D.*

Proof. Transforming $s_1 \in S$ with $\mathcal{C}(s_1, k)$ in a L-k-CNF, where $k := r$, results due to $\mathcal{L}(\cdot)$ in a Boolean formula which is only satisfiable with bit-strings from $\{0, 1\}^L \setminus F_1$, where the symbol $*$ represents either a 1 or 0 and

$$\begin{aligned}
F_1 = \Bigg\{ & s_1[1, \ldots, r] \underbrace{* * \cdots *}_{l-r}, \\
& * s_1[2, \ldots, r+1] \underbrace{* * \cdots *}_{l-r-1}, \\
& \qquad\qquad \vdots \\
& \underbrace{* * \cdots *}_{l-r} s_1[l-r+1, \ldots, l] \Bigg\}.
\end{aligned}$$

Transforming the remaining $s_i = s_2, s_3, \ldots, s_{|S|}$ with $\mathcal{C}(s_i, k)$ and constructing $\widehat{\phi}_{rcb} = \phi_{rcb}^1 \wedge \phi_{rcb}^2 \wedge \cdots \wedge \phi_{rcb}^{|S|}$ results in a Boolean formula which is only satisfiable with bit-strings from $\{0, 1\}^L \setminus (F_1 \cup F_2 \cup \cdots \cup F_{|S|})$. Each r-contiguous detector from D has *no* matching bits at $s_i[1, \ldots, r], s_i[2, \ldots, r+1], \ldots, s_i[L-r+1, \ldots, L]$ for $i = 1, 2, \ldots, |S|$. Hence, $\widehat{\phi}_{rcb}$ is only satisfiable with assignment set $\{0, 1\}^L \setminus (F_1 \cup F_2 \cup \cdots \cup F_{|S|}) = D$. □

5.4.3. Unsatisfiable CNF formula and no generable detectors

In this section, we use our obtained transformation result (Proposition 3) to demonstrate involving properties on the number of generable r-contiguous detectors. An example is the question: Given S and r, is it possible to generate any detectors at all? By obtaining with $\mathcal{C}(s, r)$ a Boolean formula $\widehat{\phi}_{rcb}$ in CNF, this question can be answered by means of the resolution method [84,85]. The resolution is a method for demonstrating that a CNF formula is unsatisfiable, i.e. a deduction to the empty clause (symbol □), or in our case that no detectors can be generated. Roughly speaking, it

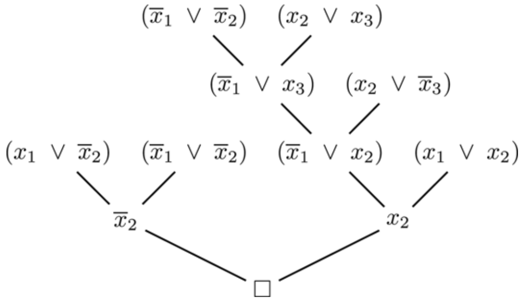
$(\overline{x}_1 \lor \overline{x}_2)$
$(x_2 \lor x_3)$
$(\overline{x}_1 \lor x_3)$
$(x_2 \lor \overline{x}_3)$
$(x_1 \lor \overline{x}_2)$
$(\overline{x}_1 \lor \overline{x}_2)$
$(\overline{x}_1 \lor x_2)$
$(x_1 \lor x_2)$
$\overline{x}_2$
x_2
$\square$

Fig. 7. Resolution method results in the empty clause $\square$ and implies that $\widehat{\phi}_{rcb}$ is not satisfiable.

is based on the idea of successively adding resolvents to the formula. Resolvents are clauses which do not modify the (growing) formula.

Specifically, let C_i and C_j be clauses and let x be a literal which occurs in C_i and also occurs in C_j as $\overline{x}$, i.e. $x \in C_i$ and $\overline{x} \in C_j$. The resolvent of C_i and C_j is $C'_i \cup C'_j$, where $C'_i := C_i \setminus \{x\}$ and $C'_j := C_j \setminus \{\overline{x}\}$. For example, $(x_1 \lor x_3)$ is the resolvent of $(x_1 \lor x_2)$ and $(x_1 \lor \overline{x}_2 \lor x_3)$.

Example 1. Let S contain the following bit-strings $\{110, 000, 010, 001\}$ and let $r = 2$. The obtained Boolean formula $\widehat{\phi}_{rcb}$ results in

$$\widehat{\phi}_{rcb} = (\overline{x}_1 \lor \overline{x}_2) \land (\overline{x}_2 \lor x_3) \land (x_1 \lor x_2) \land (x_2 \lor x_3) \land (x_1 \lor \overline{x}_2) \land (\overline{x}_2 \lor x_3) \\ \land (x_1 \lor x_2) \land (x_2 \lor \overline{x}_3).$$

By applying the resolution method (see Fig. 7), one can see that $\widehat{\phi}_{rcb}$ is reduced to the empty clause $\square$, i.e. $\widehat{\phi}_{rcb}$ is *not* satisfiable and therefore no detectors are generable.

However, we would like to emphasize here that the resolution method for determining if detectors are generable is interesting mainly from the theoretical point of view. Unfortunately, it takes an exponential number of resolution steps until an empty clause is obtained. Information on the complexity of the resolution method is provided in [85].

Another approach to answer the question: Is it possible to generate any detectors at all? is to apply a variant of the Lovász Local Lemma [85]. More specifically we define according to [85], $vbl(C)$ as the set of variables that occur in clause C, i.e. $\{x \in V | x \in C \text{ or } \overline{x} \in C\}$, where V is a set of Boolean variables. Moreover, as defined in [85], the *neighborhood* of C in ϕ_{rcb} is the set of clauses distinct from C in ϕ_{rcb} that depend on C, or more formally:

$$\Gamma_{\phi_{rcb}}(C) := \{C' \in \phi_{rcb} \,|\, C' \neq C \text{ and } vbl(C) \cap vbl(C') \neq \emptyset\}$$

Proposition 4. *Let S be a set of bit-strings of length L, where all $s \in S$ are consisting of pairwise distinct substrings $s[1, \ldots, r], s[2, \ldots, r+1], \ldots, s[L-r+1, \ldots, L]$. R-contiguous detectors are generable, if*

$$|S| < \frac{2^r\, e^{-1} + 1}{2r - 1}.$$

Proof. For each $s \in S$ construct a Boolean formula ϕ^i_{rcb} in L-k-CNF by $\mathcal{C}(s, r)$. Construct a related k-CNF Boolean formula $\widehat{\phi}_{rcb} = \phi^1_{rcb} \land \phi^2_{rcb} \land \cdots \land \phi^{|S|}_{rcb}$. Let C^j_i be the i-th clause in ϕ^j_{rcb}, $1 \leq j \leq |S|$. C^j_i has at most $2\,(r-1)$ many neighborhood clauses in ϕ^j_{rcb} and at most $(2\,(r-1)+1) \cdot (|S|-1)$ many neighborhood clauses in all remaining Boolean formulas $\phi^1_{rcb}, \phi^2_{rcb}, \ldots, \phi^{j-1}_{rcb}, \phi^{j+1}_{rcb}, \ldots, \phi^{|S|}_{rcb}$. In total this results in $|S| \cdot (2r-1) - 1$ dependent clauses (see Fig. 8).

A variant of the Lovász Local Lemma [85] implies that if $|\Gamma_F(C)| \leq 2^{k-2},\ k \in \mathbb{N}$ for all clauses C in a k-CNF formula F, then F is satisfiable. Applying the variant of the Lovász Local Lemma results in

$$|S| \cdot (2r-1) - 1 \leq 2^{r-2} < 2^r/e. \quad \square$$

$$\begin{array}{ll}
\phi_{rcb}^{1} & =(x_1 \vee x_2 \vee \ldots \vee x_r) \wedge (x_2 \vee x_3 \vee \ldots \vee x_{r+1}) \wedge \ldots \wedge (x_i \vee x_{i+1} \vee \ldots \vee x_{i+r+1}) \wedge \ldots \wedge (x_{L-r+1} \vee x_{L-r+2} \vee \ldots \vee x_L) \\
\phi_{rcb}^{2} & =(x_1 \vee x_2 \vee \ldots \vee x_r) \wedge (x_2 \vee x_3 \vee \ldots \vee x_{r+1}) \wedge \ldots \wedge (x_i \vee x_{i+1} \vee \ldots \vee x_{i+r+1}) \wedge \ldots \wedge (x_{L-r+1} \vee x_{L-r+2} \vee \ldots \vee x_L) \\
\vdots & \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\phi_{rcb}^{j} & =(x_1 \vee x_2 \vee \ldots \vee x_r) \wedge (x_2 \vee x_3 \vee \ldots \vee x_{r+1}) \wedge \ldots \wedge \underbrace{\overbrace{(x_i \vee x_{i+1} \vee \ldots \vee x_{i+r+1})}^{C_i^j}}_{|\Gamma_{\phi_{rcb}^{j}}(C_i^j)|=2(r-1)} \wedge \ldots \wedge (x_{L-r+1} \vee x_{L-r+2} \vee \ldots \vee x_L) \\
\vdots & \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\phi_{rcb}^{|S|} & =(x_1 \vee x_2 \vee \ldots \vee x_r) \wedge (x_2 \vee x_3 \vee \ldots \vee x_{r+1}) \wedge \ldots \wedge (x_i \vee x_{i+1} \vee \ldots \vee x_{i+r+1}) \wedge \ldots \wedge (x_{L-r+1} \vee x_{L-r+2} \vee \ldots \vee x_L)
\end{array}$$

Fig. 8. C_i^j has at most $2\,(r-1)$ many neighborhood clauses in ϕ_{rcb}^{j} ($r-1$ to left and $r-1$ to right) and at most $(2\,(r-1)+1)\cdot(|S|-1)$ many neighborhood clauses in all remaining Boolean formulas $\phi_{rcb}^{1}, \phi_{rcb}^{2}, \ldots, \phi_{rcb}^{j-1}, \phi_{rcb}^{j+1}, \ldots, \phi_{rcb}^{|S|}$.

5.4.4. *Complexity of L-k-CNF*

There seems to be strong evidence that finding *all* r-contiguous detectors requires at least $\Omega(2^k)$ bit string evaluations. This conclusion is justified by the fact that $\Omega(2^k)$ evaluations are required for finding all satisfying sets for the first clause of each $s \in S$. Additionally, the remaining $(l-r)$ clauses of each $s \in S$ must be verified, which in total could be done in at most $\mathcal{O}(|S| \cdot 2^k)$ evaluations. Moreover as outlined in Section 5.4.2, the k-CNF satisfiability problem is a decision problem, where the input is a boolean formula f and the output is "Yes", if f is satisfiable, and "No", otherwise. The currently fastest known deterministic algorithm that decides the 3-CNF problem, runs in time $\mathcal{O}(1.473^n)$ [86], where n is the number of variables. The probabilistic algorithm variant runs in time $\mathcal{O}(1.3302^n)$ [87]. For $k = 4, 5, 6$ the deterministic and probabilistic algorithms runtimes become slightly worse [88].

We would like to emphasize here that the (deterministic and probabilistic) k-CNF algorithms proposed in [88,87] decide if a boolean formula is satisfiable. However, the algorithms do not output *all* satisfiable assignment sets — in our case, all generable detectors.

6. Conclusions

We have reviewed the recent theoretical advances in artificial immune systems (AIS). Up until now, AIS algorithms have mainly been developed in an *ad hoc* manner, with particular applications in mind. This has meant that theoretical justification for the use of AIS has mostly been lacking. The theoretical work done so far merely constitutes the first steps towards developing a more rigorous underpinning to the area, and there clearly remains a great deal more work to be done. For several reasons, existing theoretical studies of AIS have predominately focussed on clonal selection and negative selection algorithms. For clonal selection-type algorithms, the comparison with genetic algorithms and other evolutionary computational approaches cannot be ignored. There are now many significant theoretical results on evolutionary algorithms (both in terms of schema [89], and using the theory of Markov chains [71]). A natural application of these results would be to immune algorithms of an evolutionary nature, since the latter have many similarities with genetic algorithms, for instance. Also, since the most prolific use of AIS algorithms has involved the clonal selection and negative selection approaches, the theoretical advances in those types of algorithms (such as those reviewed here) are likely to be of most use to the community.

We have seen that whilst convergence proofs were initially only available for specific algorithms, such as MISA [68], and the B cell algorithm [64], there is now a suitable framework for understanding the convergence of generic clonal selection algorithms [72]. Due to the fact that clonal selection is applied to cell populations in immune networks, some of this convergence theory will also be relevant to immune network algorithms. However, we have hinted that a full theory of the latter would require a proper understanding of the interplay between the nonlinear dynamics of the network and the stochastic nature of the clonal selection mechanism, which is currently lacking.

For theory to be of use to the wider community, it must be seen to give some insight into the possible application (or otherwise) of an approach. The work reviewed here on negative selection has been in this vein, with clear theoretical results showing that there are limitations with the current approaches being employed, and that the applicability of negative selection algorithms in certain areas may not be as appropriate as was first assumed. As we have seen, the problem of generating detectors in a negative selection algorithm turns out to be equivalent to an $\mathcal{NP}$-complete problem. More generally, when applying AIS algorithms to complex problems, the issue of scalability is a serious one. In [90] an optimal control-based approach is suggested for dealing with the allocation of resources when applying an AIS to solve a complex, dynamic problem.

It has been noted by some that the field of AIS has somewhat drifted away from its immunological roots [91, 92]. In this paper we have made a brief attempt to explain the immunology behind the algorithms, but the observant reader should notice that the algorithms developed are poor cousins of the immune system mechanisms from which they are derived. However, in the past few years, work by the Danger Team[4] has started to address this imbalance. For example, recent work by [93] has begun explorations into the use of *dendritic cells* (which are a type of cell found in the innate immune system), as a mechanism for identifying *dangerous* (or anomalous) events in a data stream. Whilst that work is still preliminary and works only on static data at the moment, it may go some way towards making a real breakthrough in the intrusion detection area of AIS research. Similarly, preliminary work in [94] proposes an *artificial tissue*, which is a type of representation of the data space that can evolve and adapt over time, providing a useful bridge between the data and the immune algorithm itself.

To try to create a greater synergy between immunology and computer science, the authors of [91] propose a conceptual framework that allows for the development of AIS algorithms that are more grounded in biology, through the adoption of a fully interdisciplinary approach. The biological metaphors employed in AIS research thus far have typically been simple, but somewhat effective. However, as proposed in [91], through greater interaction between computer scientists, engineers, biologists and mathematicians, better insights into the workings of the immune system and the applicability of the AIS paradigm will be gained. The development of AIS techniques should be rooted in a sound methodology in order to fully exploit the paradigm: for instance, modelling techniques such as cellular automata [95] and process calculi such as π-calculus [96], and stochastic π-calculus [97] could be employed. Thus, in order to develop an AIS algorithm appropriately, it is first necessary to develop mathematical and/or computational models of the immune system, focussing on the biological aspects of interest. Apart from the computational applications, there is great potential for AIS to make a contribution to immunology through this process. Of course, modelling immune systems is nothing new, and the field of AIS itself arose from this area to some extent. Yet there is the potential for novel results to emerge in at least two ways: first, algorithms that have been developed through this process should better exploit the underlying biological principles on which they have been based, and second, computer scientists can pose apparently naïve questions about immunology that cause immunologists to think in quite a different way, yielding new biological insights. In future, the adoption of a 'theory first' approach, starting with detailed theoretical models of the biology, will provide a sound basis for the algorithms developed.

References

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

[2] E. Hart, J. Timmis, Application areas of AIS: The past, the present and the future, Applied Soft Computing 8 (1) (2008) 191–201.

[3] E. Hart, P. Ross, Studies on the implications of shape–space models for idiotypic networks, in: Proceedings of the 3rd International Conference on Artificial Immune Systems, ICARIS 2004, in: LNCS, vol. 3239, Springer, 2004, pp. 413–426.

[4] E. Hart, Not all balls are round, in: C. Jacob, M. Pilat, P. Bentley, J. Timmis (Eds.), Proceedings of the 4th International Conference on Artificial Immune Systems, in: LNCS, vol. 3627, Springer, 2005.

[5] E. Hart, Analysis of a growth model for idiotypic networks, in: H. Bersini, J. Carneiro (Eds.), 5th International Conference on Artificial Immune Systems, ICARIS, vol. 4163, 2006, pp. 66–80.

[6] E. Hart, H. Bersini, F. Santos, Tolerance vs intolerance: How affinity defines topology in an idiotypic network, in: H. Bersini, J. Carneiro (Eds.), 5th International Conference on Artificial Immune Systems, ICARIS, vol. 4163, 2006, pp. 109–121.

[7] I.R. Cohen, Tending Adam's Garden: Evolving the Cognitive Immune Self, Elsevier Academic Press, 2000.

[8] C.A. Janeway Jr., R. Medzhitov, Innate immune recognition, Annual Review of Immunology 20 (2002) 197–216.

[9] Charles A. Janeway Jr., Travers Paul, Immunobiology: The Immune System in Health and Disease, third ed., Garland Publishing, New York, 1997.

[10] R.N. Germain, An innately interesting decade of research in immunology, Nature Medicine 10 (12) (2004) 1307–1320.

[11] F.M. Burnet, The Clonal Selection Theory of Acquired Immunity, Cambridge University Press, 1959.

[4] http://www.dangertheory.com.

[12] S. Forrest, A.S. Perelson, L. Allen, R. Cherukuri, Self-nonself discrimination in a computer, in: IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press, Los Alamos, CA, 1994.

[13] F. Esponda, Negative representations of information, Ph.D. Thesis, University of New Mexico, 2005.

[14] N.K. Jerne, Towards a network theory of the immune system, Ann. Immunol. (Inst. Pasteur) 125C (1974) 373–389.

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

[16] A.S. Perelson, Immune network theory, Immunological Reviews 110 (1989) 5–36.

[17] R.E. Langman, M. Cohn, The 'complete' idiotype network is an absurd immune system, Immunology Today 7 (4) (1986) 100–101.

[18] F. Varela, A. Coutinho, B. Dupire, N. Vaz, Cognitive networks: Immune, neural and otherwise, Theoretical Immunology 2 (1988) 359–375.

[19] J.H. Holland, Adaptation in Natural and Artificial Systems, The University of Michigan Press, Ann Arbor, MI, 1975.

[20] H. Bersini, F.J. Varela, Hints for adaptive problem solving gleaned from immune network, in: Proc. 1st Workshop on Parallel Problem Solving from Nature, in: LNCS, vol. 496, Springer-Verlag, 1990, pp. 343–354.

[21] H. Bersini, F. Varela, The Immune Learning Mechansims: Recruitment, Reinforcement and their Applications, Chapman Hall, 1994.

[22] S. Forrest, A.S. Perelson, L. Allen, R. Cherukuri, Self–nonself discrimination in a computer, in: Proc. IEEE Symposium on Research Security and Privacy, 1994, pp. 202–212.

[23] R.R. Hightower, S.A. Forrest, A.S. Perelson, The evolution of emergent organization in immune system gene libraries, in: Proceedings of the 6th International Conference on Genetic Algorithms, Morgan Kaufmann, 1995, pp. 344–350.

[24] S. Forrest, S. Hofmeyr, A. Somayaji, Computer Immunology, Communications of the ACM 40 (10) (1997) 88–96.

[25] S. Hofmeyr, S. Forrest, Architecture for an artificial immune system, Evolutionary Computation 7 (1) (2000) 1289–1296.

[26] Y. Ishida, Fully distributed diagnosis by pdp learning algorithm: Towards immune network pdp model, in: Proc. of the Int. Joint Conf. on Neural Networks, 1990, pp. 777–782.

[27] Y. Ishida, An immune network model and its applications to process diagnosis, Systems and Computers in Japan 24 (6) (1993) 646–651.

[28] Y. Ishida, Active diagnosis by self-organisation: An approach by the immune network metaphor, in: Proceedings of the International Joint Conference on Artificial Intelligence, IEEE, Nagoya, Japan, 1997, pp. 1084–1089.

[29] Y. Ishida, Immunity-Based Systems: A Design Perspective, Springer-Verlag, Heidelberg, Germany, 2004.

[30] D. Cooke, J. Hunt, Recognising promoter sequences using an Artificial Immune System, in: Proceedings of Intelligent Systems in Molecular Biology, AAAI Press, 1995, pp. 89–97.

[31] J. Hunt, D. Cooke, Learning using an artificial immune system, Journal of Network and Computer Applications 19 (1996) 189–212.

[32] J. Hunt, J. Timmis, D. Cooke, M. Neal, C. King, JISYS: Development of an artificial immune system for real-world applications, in: D. Dasgupta (Ed.), Artificial Immune Systems and their Applications, Springer, 1998, pp. 157–186.

[33] J. Timmis, Artificial immune systems: A novel data analysis technique inspired by the immune system, Ph.D. Thesis, University of Wales, Aberystwyth, 2000.

[34] L.N. de Castro, F.J. Von Zuben, Learning and optimization using the clonal selection principle, IEEE Transactions on Evolutionary Computation 6 (3) (2002) 239–251.

[35] L.N. de Castro, F.J. Von Zuben, aiNet: An Artificial Immune Network for Data Analysis, Idea Group Publishing, USA, 2001, 231–259.

[36] S. Wierzchon, U. Kuzelewska, Stable clusters formation in an artificial immune system, in: Jonathan Timmis, Peter J. Bentley (Eds.), Proceedings of the 1st International Conference on Artificial Immune Systems, ICARIS, University of Kent at Canterbury, September 2002, University of Kent at Canterbury Printing Unit, 2002, pp. 68–75.

[37] M. Neal, An artificial immune system for continuous analysis of time-varying data, in: Jonathan Timmis, Peter J. Bentley (Eds.), Proceedings of the 1st International Conference on Artificial Immune Systems, ICARIS, University of Kent at Canterbury, September 2002, University of Kent at Canterbury Printing Unit, 2002, pp. 76–85.

[38] E. Hart, P. Ross, Exploiting the analogy between immunology and sparse distributed memories: A system for clustering non-stationary data, in: Jonathan Timmis, Peter J. Bentley (Eds.), Proceedings of the 1st International Conference on Artificial Immune Systems, ICARIS, University of Kent at Canterbury, September 2002, University of Kent at Canterbury Printing Unit, 2002, pp. 49–58.

[39] A. Watkins, An artificial immune recognition system, M.Sc. Thesis, Mississippi State University, 2001.

[40] A. Watkins, J. Timmis, L. Boggess, Artificial immune recognition system (AIRS): An immune inspired supervised machine learning algorithm, Genetic Programming and Evolvable Machines 5 (3) (2004) 291–318.

[41] D. Goodman, L. Boggess, A. Watkins, An investigation into the source of power for airs, an artificial immune classification system, in: Proc. of International Joint Conference on Neural Networks, IEEE, 2003, pp. 1678–1683.

[42] D. Goodman, L. Boggess, A. Watkins, Artificial immune system classification of multiple-class problems, in: Proc. of Intelligent Engineering Systems, ASME, 2002, pp. 179–184.

[43] J. Greensmoth, New frontiers for an artificial immune system, M.Sc. Thesis, School of Computing, University of Leeds, 2003.

[44] A. Watkins, Exploiting immunological metaphors in the development of serial, parallel and distributed learning algorithms, Ph.D. Thesis, University of Kent, Computing Laboratory, 2005.

[45] D. Dasgupta, An overview of artificial immune systems and their applications, in: Artificial Immune Systems and their Applications, Springer-Verlag, Inc., 1999, pp. 1–21 (Chapter 1).

[46] J. Timmis, T. Knight, Artificial immune systems: Using the immune system as inspiration for data mining, in: H. Abbas, A. Ruhul, A. Sarker, S. Newton (Eds.), Data Mining: A Heuristic Approach, Idea Group, 2001, pp. 209–230.

[47] S. Garrett, How do we evaluate artificial immune systems?, Evolutionary Computation 13 (2) (2005) 145–177.

[48] J. Timmis, P. Bentley, Proc. of the 1st International Conference on Artificial Immune Systems, University of Kent Printing Unit, 2002.

[49] J. Timmis, P. Bentley, E. Hart, Proc. of the 2nd International Conference on Artificial Immune Systems, ICARIS, in: Lecture Notes in Computer Science, vol. 2787, Springer, 2003.

[50] G. Nicosia, V. Cutello, P. Bentley, J. Timmis, Proc. of the 3rd International Conference on Artificial Immune Systems, ICARIS, in: Lecture Notes in Computer Science, vol. 3239, Springer, 2004.

[51] C. Jacob, M. Pilat, P. Bentley, J. Timmis, Proc. of the 4th International Conference on Artificial Immune Systems, ICARIS, in: Lecture Notes in Computer Science, vol. 3627, Springer, 2005.

[52] H. Bersini, J. Carneiro (Eds.), Proceedings of the 5th International Conference on Artificial Immune Systems, vol. 4163, Springer, 2006.

[53] L.N. de Castro, F.J. Von Zuben, H. Knidel (Eds.), Proceedings of the 6th International Conference on Artificial Immune Systems, in: Lecture Notes in Computer Science, vol. 4628, Springer, 2007.

[54] A. Freitas, J. Timmis, Revisiting the foundations of artificial immune systems for data mining, IEEE Transactions on Evolutionary Computation 11 (4) (2007) 521–540.

[55] A.S. Perelson, G.F. Oster, Theoretical studies of clonal selection: Minimal antibody repertoire size and reliability of self-nonself discrimination, 81 (1979) 645–670.

[56] L.A. Segal, A.S. Perelson, Computations in shape–space: A new approach to immune network theory, in: Theoretical Immunology, Part Two, 1988.

[57] Alan S. Perelson, G. Weisbuch, Immunology for physicists, Reviews of Modern Physics 69 (4) (1997) 1219–1267.

[58] J.K. Percus, O.E. Percus, A.S. Perelson, Predicting the size of the T-cell receptor and antibody combining region from consideration of efficient self-nonself discrimination, Proceedings of National Academy of Sciences USA 90 (1993) 1691–1695.

[59] G. Nicosia, Immune algorithms for optimisation and protein structure prediction, Ph.D. Thesis, Department of Mathematics and Computer Science, University of Catania, Italy, 2004.

[60] J. Kelsey, J. Timmis, Immune inspired somatic contiguous hypermutation for function optimisation, in: Genetic and Evolutionary Computation Conference, in: Lecture Notes in Computer Science, vol. 2723, Springer, 2003, pp. 207–218.

[61] C.A. Coello Coello, N. Cruz Cortes, An approach to solve multiobjective optimization problems based on an artificial immune system, in: Jonathan Timmis, Peter J. Bentley (Eds.), Proceedings of the 1st International Conference on Artificial Immune Systems, ICARIS, University of Kent at Canterbury, September 2002, pp. 212–221, University of Kent at Canterbury Printing Unit, 2002.

[62] N. Cruz Cortes, C. Coello Coello, Multiobjective optimisation using ideas from the clonal selection principle, in: E. Cantu-Paz (Ed.), Genetic and Evolutionary Computation (GECCO), vol. 1, 2003, pp. 158–170.

[63] M. Mitchell, An Introduction to Genetic Algorithms, 6th ed., The MIT Press, 1999.

[64] E. Clark, A. Hone, J. Timmis, A Markov chain model of the B-cell algorithm, in: C. Jacob, M. Pilat, P. Bentley, J. Timmis (Eds.), Proceedings of the 4th International Conference on Artificial Immune Systems, in: LNCS, vol. 3627, Springer, 2005, pp. 318–330.

[65] L.N. de Castro, F.J. von Zuben, An evolutionary immune network for data clustering, in: Proceedings of the 6th Brazilian Symposium on Neural Networks, IEEE Computer Society, 2000, pp. 84–89.

[66] L.N. de Castro, F.J. von Zuben, aiNet: An artificial immune network for data analysis, in: Hussein A. Abbass, Ruhul A. Sarker, Charles S. Newton (Eds.), Data Mining: A Heuristic Approach, Idea Group Publishing, 2001, pp. 231–259 (Chapter 12).

[67] G.R. Grimmett, D.R. Stirzaker, Probability and Random Processes, Oxford University Press, 1982.

[68] M. Villalobos-Arias, C.A. Coello Coello, O. Hernandez-Lerma, Convergence analysis of a multiobjective artificial immune system algorithm, Lecture Notes in Computer Science 3239 (2004) 226–235.

[69] A.E. Nix, M.D. Vose, Modeling genetic algorithms with markov chains, Annals of Mathematics and Artificial Intelligence 5 (1992) 79–88.

[70] M.D. Vose, Modeling simple genetic algorithms, in: Evolutionary Computation, vol. 3, 1996.

[71] G. Rudolph, Finite markov chain results in evolutionary computation: A tour d'horizon, Fundamenta Informaticae 35 (1–4) (1998) 67–89.

[72] V. Cutello, G. Nicosia, P. Oliveto, M. Romeo, On the convergence of immune algorithms, in: Proc.of Foundations of Computational Intelligence, 2007, pp. 409–416.

[73] E. Seneta, Non-Negative Matrices and Markov Chains, Springer-Verlag, New York, 1981.

[74] F. Robert, Les Systèmes Dynamiques Discrets, Springer-Verlag, Berlin, 1995.

[75] D.W. Jordan, P. Smith, Nonlinear Ordinary Differential Equations, third ed., Oxford University Press, Oxford, 1999.

[76] S. Elaydi, Discrete Chaos, Chapman and Hall/CRC, Boca Raton, 2000.

[77] A. Hone, J. Kelsey, Optima, extrema and artificial immune systems, in: Lecture Notes in Computer Science, Springer, 2004, pp. 89–98.

[78] Larisa B. Goncharova, Alexander O. Tarakanov, Yuri Melnikov, Biomolecular immunocomputing, in: Proceedings of the 2nd International Conference on Artificial Immune Systems, ICARIS 2003, in: LNCS, vol. 2787, Springer, 2003, pp. 102–110.

[79] A.O. Tarakanov, V.A. Skormin, S.P. Sokolova, Immunocomputing: Principles and Applications, Springer-Verlag, New York, NY, 2003.

[80] William Feller, An Introduction to Probability Theory and its Applications, 3rd ed., vol. 1, John Wiley & Sons, 1968.

[81] M.T. Ranang, An artificial immune system approach to preserving security in computer networks, Master's Thesis, Norges Teknisk-Naturvitenskapelige Universitet, 2002.

[82] James Victor Uspensky, Introduction to Mathematical Probability, McGraw-Hill, 1937.

[83] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, second ed., MIT Press, 2002.

[84] J.A. Robinson, A machine-oriented logic based on the resolution principle, Journal of the Association for Computing Machinery (JACM) 12 (1) (1965) 23–41.

[85] E. Welzl, Boolean satisfiability — combinatorics and algorithms, in: Lecture Notes, 2005, http://www.inf.ethz.ch/~emo/SmallPieces/SAT.ps.

[86] T. Brueggemann, W. Kern, An improved deterministic local search algorithm for 3-SAT, Theoretical Computer Science 329 (1–3) (2004) 303–313.

[87] T. Hofmeister, U. Schöning, R. Schuler, O. Watanabe, A probabilistic 3-SAT algorithm further improved, in: 19th Annual Symposium on Theoretical Aspects of Computer Science, STACS, in: Lecture Notes in Computer Science, vol. 2285, Springer-Verlag, 2002, pp. 192–202.

[88] U. Schöning, A probabilistic algorithm for k-SAT and constraint satisfaction problems, in: 40th Annual Symposium on Foundations of Computer Science, FOCS, IEEE Press, 1999, pp. 410–414.

[89] W.B. Langdon, R. Poli, Foundations of Genetic Programming, Springer-Verlag, Berlin, 2002.

[90] A. Hone, H. van den Berg, Mathematical analysis of artificial immune system dynamics and performance, in: D. Flower, J. Timmis (Eds.), Silico Immunology, Springer, 2007, pp. 351–374.

[91] S. Stepney, R. Smith, J. Timmis, A. Tyrrell, M. Neal, A. Hone, Conceptual frameworks for artificial immune systems, International Journal of Unconventional Computing 1 (3) (2005) 315–338.

[92] J. Timmis, Artificial immune systems: Today and tommorow, Natural Computing 6 (1) (2007).

[93] J. Greensmith, U. Aickelin, J. Twycross, Articulation and clarification of the dendritic cell algorithm, in: H. Bersini, J. Carneiro (Eds.), Proceedings of the 5th International Conference on Artificial Immune Systems, vol. 4163, 2006, pp. 404–417.

[94] P.J. Bentley, J. Greensmith, S. Ujjin, Two ways to grow tissue for Artificial Immune Systems, In Jacob et al. [51], pp. 139–152.

[95] H.B. Sieburg, O.K. Clay, The cellular device machine development system for modeling biology on the computer, Complex Systems 5 (1991) 575–601.

[96] C. Kuttler, J. Niehren, R. Blossey, Gene regulation in the pi calculus: Simulating cooperativity at the lambda switch, in: Proc Concurrent Models in Molecular Biology, Bioconcur 04, 2004.

[97] A. Phillips, L. Cardelli, A correct abstract machine for the stochastic pi-calculus, in: Proc Concurrent Models in Molecular Biology, Bioconcur'04, ENTCS, 2004.