
How Do We Evaluate Artificial Immune Systems?

Simon M. Garrett

smg@aber.ac.uk

Computational Biology Group, Department of Computer Science,
University of Wales, Aberystwyth, Wales, SY23 3DB. UK

Abstract

The field of Artificial Immune Systems (AIS) concerns the study and development of computationally interesting abstractions of the immune system. This survey tracks the development of AIS since its inception, and then attempts to make an assessment of its usefulness, defined in terms of 'distinctiveness' and 'effectiveness.' In this paper, the standard types of AIS are examined—Negative Selection, Clonal Selection and Immune Networks—as well as a new breed of AIS, based on the immunological 'danger theory.' The paper concludes that all types of AIS largely satisfy the criteria outlined for being useful, but only two types of AIS satisfy both criteria with any certainty.

Keywords

Artificial immune systems, critical evaluation, negative selection, clonal selection, immune network models, danger theory.

1 Introduction

What is an artificial immune system (AIS)? One answer is that an AIS is a model of the immune system that can be used by immunologists for explanation, experimentation and prediction activities that would be difficult or impossible in 'wet-lab' experiments. This is also known as 'computational immunology.' Another answer is that an AIS is an abstraction of one or more immunological processes. Since these processes protect us on a daily basis, from the ever-changing onslaught of biological and biochemical entities that seek to prosper at our expense, it is reasoned that they may be computationally useful. It is this form of AIS—methods based on immune abstractions—that will be studied here.

Although AIS is a relatively young field, advancing on many fronts, some central themes have become apparent—the question is, are these AIS delivering anything *useful*, or are they just another addition to the increasingly long line of approaches that are biologically inspired? These approaches include established paradigms such as genetic and evolutionary computation (GEC), artificial neural networks (ANN) and various forms of artificial life; as well as less established topics such as ant colony dynamics (Dorigo, 1992; Dorigo, 1999) and cell membrane computing (Paun, 2002). The intention here is to provide an assessment of prior developments in AIS, its current strengths, weaknesses and its overall usefulness.

There have been several surveys of AIS, and a few comparisons between AIS and other methods, e.g. (Dasgupta, 1997; Dasgupta, 1999; Perelson and Weisbuch, 1997; ?; ?). Unfortunately, most of this work is now somewhat out-of-date; only Perelson has reviewed three of the main themes of AIS, but did so from an immunological point of

view; none have discussed danger theory, and there has been no specific attempt to assess the *usefulness* of AIS, with a view to its future development, which is the central focus here. It would appear that answers to the following questions are of value as an introduction and critique of AIS, and its relationship to other paradigms:

- How have the various types of AIS developed, and what are their positive and negative features?
- What are the criteria for deciding whether methods, such as AIS, are useful?
- If ‘distinctiveness’ is one criterion of usefulness, how distinct are AIS from other paradigms, such as GEC and ANN?
- If ‘effectiveness’ is another criterion of usefulness, what can AIS do uniquely, better or faster than other methods?
- Having applied the assessment criteria, are AIS useful? What does this suggest for the future development of AIS?

The paper is organized as follows: Section 2 argues that a computational method is ‘useful’ when it is both *distinct* and highly *effective*, and it explores what is meant by these terms. Sections 3 to 6 provide a survey of the most important types of AIS, namely Negative Selection, Danger Models, Clonal Selection and Immune Network Models. Each section explores how one type of AIS has been developed to the current state-of-the-art; lists its key applications, and applies the ‘usefulness criteria’ as defined in Section 2. Section 7 summarizes the results of applying the usefulness criteria to each type of AIS (for ease of comparison) and, in light of these assessments, answers the question, ‘Are AIS useful?’

2 Defining ‘Useful’ in Terms of ‘Distinct’ and ‘Effective’

An algorithm may be distinctly different from other algorithms but ineffective, or it may be highly effective but be reducible to other, existing paradigms, and therefore lacking in distinctiveness. However, if a method is both distinct and effective, then it offers a truly *useful* means of computation. These criteria of ‘distinctiveness’ and ‘effectiveness’ will be used for assessing the usefulness of AIS. They may also prove to be relevant in testing the usefulness of other computational methods.

The terms ‘distinctiveness’ and ‘effectiveness’ will be further defined; however, due to the imprecision of language, and the difficulties of comparing any two non-trivial systems, these two criteria can only ever hope to be somewhat blunt instruments for assessing usefulness. The definitions and criteria given here for the words ‘useful’, ‘distinct’ and ‘effective’ may not agree with the reader’s, or anyone else’s for that matter, but they can provide an approximation of ‘usefulness’ when applied consistently.

What Makes a Research Paradigm Distinctive? With a biologically inspired paradigm, such as AIS, there may be a temptation to appeal to its source of inspiration as an indication of its distinctiveness; this appears to be a mistake. Firstly, it is possible to envisage methods that have different sources of inspiration, but which result in mathematically identical methods. Secondly, a single source of inspiration, such as immunology, has given rise to several types of AIS. Thirdly, even biologically implausible, or unlikely, ideas can inspire distinct mathematical algorithms. The source of inspiration is not a reliable test for distinctiveness.

A less ambiguous way to assess how distinct one method is from another is to convert both methods to a shared, common language. Algorithmic or mathematical expressions appear to be an ideal formalism that allow reliable comparisons to be made, and such an approach also helps avoid anthropomorphisms (or immunomorphisms!) as highlighted by (McDermott, 1981). The problem is that the comparison process is often non-trivial. For example (Smith et al., 1996) and (Hart and Ross, 2002) discuss the intricate relationship between AIS, Sparse Distributed Memory and ANNs. Even comparing two AIS methods that are closely related, such as clonal selection and immune network models, requires careful thought.

The following test questions are suggested, which somewhat relate to the definition of the Physical Symbol System (PSS) (Newell and Simon, 1976), taking just the main features of the PSS; namely a set of *symbols* that are organized into *expressions*, operated on by *processes* that change those expressions over time. Given a supposed new method, the test questions are:

- D.1 Does the new method contain unique *symbols*, or can the features of this method be transformed into the features of the another method, without affecting the dynamics of the new method?
- D.2 Are the new method's symbols organized in novel *expressions*, or can its expressions be transformed to become the same as some other method, without affecting its dynamics?
- D.3 Does the new method contain unique *processes* that are applied to its expressions, or can its processes be transformed to become identical to some other method, without affecting its dynamics?

If the answer is 'no' to all these questions then there is almost certainly nothing distinctive in the supposedly new method. The more questions that can be answered 'yes', the more likely that the method is distinct in some way from the existing method.

No single researcher can claim to have tested their 'new method' in this way against every other existing method—this duty also falls to members of the wider academic community, who can attempt to falsify any assertion of distinctiveness made about the new method. However, it is *not* enough to state, for example, that an AIS method is indistinguishable from GEC or ANN, because it contains a mutation operator, or uses the concept of a network of entities. Many systems will partially share symbols, expressions and/or processes, but methods can be said to be distinct if their symbols, expressions and processes *as a whole* can not be made equivalent.

The outcome of successfully passing the tests above is that the new method truly adds to the choice of methods that were previously available. For example, if one is choosing which method to use for automatically generating a robot controller, what matters is the mathematically predictable performance of the methods. *Iff* a method is distinct from other methods then it adds to this choice of methods.

What Makes a Research Paradigm Effective? A similar set of questions can help when assessing the effectiveness of an AIS method, although they are more obvious because assessing effectiveness is standard scientific practice:

- E.1 Does the method provide a *unique* means of obtaining a set of results?
- E.2 Does the method provide *better* results than existing methods, when applied to a shared benchmark task?

E.3 Does the method allow a set of results to be obtained *more quickly* than another method, on a benchmark test? The same types of experiments should be run in both cases: there is no point, for example, in comparing the average best fitness in one test with the best fitness in another test.

For a method to be effective, the answer must be ‘yes’ to at least one of these questions; unfortunately there is a complication with AIS methods. Since at least some of the AIS methods described below can be reduced to search algorithms, the No Free Lunch (NFL) theorem applies (Wolpert and Macready, 1995); i.e. no method is more effective than any other when compared over all possible functions. The questions E.1–E.3 are valid, however, over a restricted class of tasks; therefore, a method will be regarded as ‘effective’ if there is a positive answer to one of the questions above, *and* when the method is applied to a well-defined subset of applications or function tasks. Where the NFL theorem does not apply to an AIS algorithm, points E.1 to E.3 are still important; however, E.3 becomes easier to demonstrate.

Finally, the NFL theorem says that *every* search algorithm will excel in at least one subset of search problem, so there is an additional requirement: ‘Is this subset of problems of practical interest?’ In conclusion, if an AIS method is better, faster or unique in solving tasks of practical interest, then it will be regarded as effective.

Combining Distinctiveness and Effectiveness Since distinctiveness is assessed by *static*, formal analysis of an algorithm, relative to one or more other algorithms, and effectiveness is measured by their *active* relative performance in experiments, there is likely to be a certain amount of redundancy when using both criteria. For example, two indistinct methods will have identical effectiveness, and two equally effective methods *may* be reducible to a shared algorithmic form. Nevertheless, due to the difficulties of applying the criteria outlined above, it is prudent to be over-cautious and require that an AIS method must be both effective and distinct before it can be said to be useful.

If a method is not useful, this does not mean it should not be researched further. In 1969, ANNs were somewhat lacking in effectiveness, as famously pointed out by (Minsky and Papert, 1969), but research (eventually) continued and produced the back-propagation method, which is one of the most effective AI methods reported.

Overall, the focus here is on these practical aspects of comparing methods, not on a theoretically complete analysis. In the next four sections, these *usefulness criteria* will be applied to Negative Selection, Danger Models, Clonal Selection and Immune Network Models in turn, after a description of their development and current state-of-the-art.

3 Artificial Negative (and Positive) Selection

3.1 Background Immunology

There is significant debate about the nature of the immunological mechanism that distinguishes between an organism’s ‘self’ molecules and cells and an invading ‘nonself’ entity (Medzhitov and Janeway, 2002; ?), and indeed some doubt this dichotomy exists at all (Bersini, 2002). However, let us assume here that there is an immunological mechanism for self-nonself classification, and focus on an abstraction of one aspect of that mechanism called *negative selection*. The principle of *positive selection* has also been studied, but its symbols, expressions and processes are very similar to those of negative selection so it will only be specifically mentioned where required.

The biological negative selection of *T-cells* in the thymus occurs when T-cells that attack ‘self’ are eliminated, and only T-cells that respond weakly to self, if at all, are

released from the thymus (Medzhitov and Janeway, 2002). These cells then respond, in various degrees, to the different types of invading ‘nonself’ that need to be removed from the bloodstream, lymph, tissue, etc. Often in the AIS literature, this process is called *virus detection* but essentially the process of negative selection should provide a set of T-cell *detectors* that can detect any change in ‘self’, or any form of nonself (viruses or otherwise). It is this process that has been abstracted to form an algorithm for the detection of change in a predefined set of objects.

3.2 The Development of Artificial Negative Selection

Early Work Negative selection¹ was introduced in 1994 (Forrest et al., 1994), and is now often called *negative detection*. It is a loose abstract model of biological negative selection that concentrates on the generation of change detectors. These detectors are intended to detect when elements of a set of self strings have changed from an established norm. The algorithm of (Forrest et al., 1994) is as follows, in which they assume all strings are binary,

1. Create a set of self strings, S , by some means (e.g. see (Forrest et al., 1994).)
2. Create a set of randomly generated strings, R_0 .
3. For each $r_0 \in R_0$, form a set, R , of those r_0 that do not strongly match any $s \in S$. A strong match is defined by a matching function $m(r_0, s)$, such that: (i) $m(r_0, s) \bowtie \theta$; (ii) \bowtie is an operator, such as \geq , that defines whether high or low value of $m(r_0, s)$ indicates greater similarity between the strings, and (iii) θ defines a threshold. An example of the use of $m(x, y)$ will be given below.
4. For each r in R , ensure that no $s \in S$ matches above (or ‘below’, depending on the form of \bowtie) the threshold.
5. Return to step 4 while change detection of S is required.

Steps 1, 2 and 3 form the *censoring stage* of the algorithm, and steps 4 and 5 form the *protection stage*. The R_0 strings are censored until the desired repertoire, R , of detector strings is obtained. Note that all Forrest et al’s original papers speak of ‘ r -contiguous bits’, whereas we have ‘ k -contiguous bits’. This is because we want to use ‘ r ’ to help define the members of the set ‘ R ’.

In (Forrest et al., 1994), the matching function was the k -contiguous bits function, such that $m(r_0, s)$ returned the largest number of contiguous, matching bits, \bowtie was \geq , and $\theta = k$. In later work another matching function was used, as described below. Wierzchoń has pointed out that even a simple thresholded Hamming distance rule, $m_H(x, y)$, would be suitable (Wierzchoń, 2000); however, whereas Hamming distance is irreflexive, and $m_H(x, x) = 0$, requiring \bowtie to be \leq , k -contiguous bits distance is reflexive, and $m_C(x, x) = |x|$. For example, the k -contiguous bits measure operates as follows,

```

01101001
10101010
---- <-- four matching bits,
      so m(r_0,s) = 4.
```

¹In AIS research the adjective ‘artificial’ is dropped when discussing any type of AIS. We will also adopt this convenience.

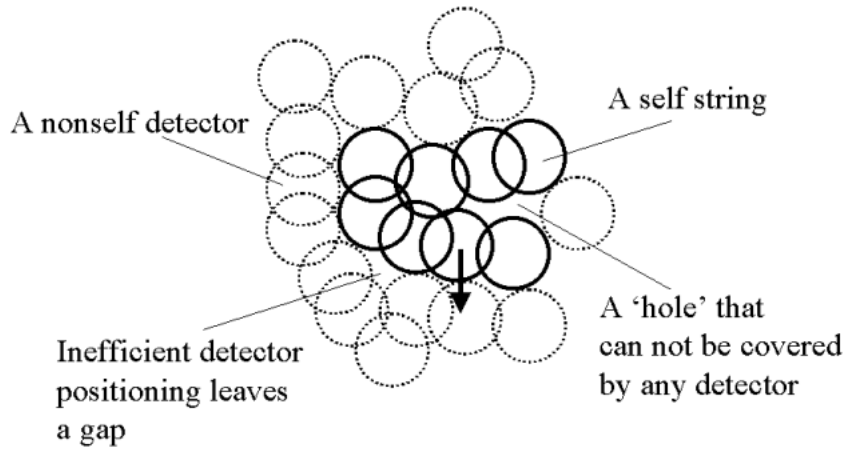


Figure 1: The principle of self protection: each nonsself detector, which need not be (hyper)spherical or circular as shown here, covers an area of the state space outside the desired self state space. If self should change, e.g. as indicated by the arrow, the change will ideally overlap one of the nonsself detectors, triggering a detection event.

The repertoire subset, R , resulting from application of the algorithm, is also known as the *detector set* since it contains elements that can detect dissimilarities to self. If self has changed it is likely that it will have become more similar to a nonsself string, increasing the match with that string, and registering as a detection event (see Figure 1.) It is also possible to create detectors to cover the self strings; this is called *positive selection* or *positive detection* and, then, detection occurs when a self element leaves a detector.

One important measure of the negative selection algorithm is the probability of failing to detect a change to self. This is related to the probability that two random strings will match, which can be stated as,

$$P_M \approx m^{-k}[(l - k)(m - 1)/m + 1], \quad (1)$$

where m = the number of symbols available for choice at any locus on the string, l = the number of symbols in a string (i.e. its length), and k = the number of contiguous matches required for a match (Percus et al., 1993). Then the probability that $|R|$ detectors will fail to detect an intrusion is,

$$P_f = (1 - P_M)^{|R|}. \quad (2)$$

Problems of Scalability and Coverage Despite this successful abstraction of negative selection, further analysis showed that there was a scalability problem. The number of random strings, $|R_0|$, grows exponentially with $|S|$, if P_f and P_M are fixed in the equation,

$$|R_0| = \frac{-\ln P_f}{P_M \times (1 - P_M)^{|S|}}. \quad (3)$$

Somewhat related to the required improvements in scalability, negative detection was inefficient in its choice of detectors. Some parts of the state space might be covered by overlapping detectors; other parts might not be covered at all. Worse still, it could be shown that certain sets of self strings would induce holes that *could not* be covered

by any nonself string(s) without also covering a self string. Improvements in detector coverage algorithms were required (see Figure 1).

Analyzing the Scalability and Coverage Problems Two suggestions for more scalable detector production were made by D'haeseleer, namely the 'linear' and 'greedy' algorithms (D'haeseleer et al., 1996).

The linear algorithm works for specific matching rules, and again assumes the use of the k -contiguous bits matching function; it works in two phases. Phase 1 solves a counting recurrence for the number of strings that are not matched by S . Phase 2 randomly picks detectors from this enumerated counting recurrence. In other words, the full possible detector set R_0 can be calculated directly, instead of by wasteful random generation, improving scalability. Then the detector set, R , can be randomly chosen from this set, in linear time, although with high space complexity.

The greedy algorithm generates detectors in a similar manner but in Phase 2 it attempts to improve the efficiency of detector coverage by choosing detectors that are as far apart as possible (D'haeseleer, 1996). This makes it slower than the linear algorithm. The reader is referred to (D'haeseleer et al., 1996) for further details of these analyses.

In addition to his work in scalability, D'haeseleer's work in detector coverage identified a serious problem, "*Even though the ... [negative detection] ... algorithm is capable of constructing a complete detector repertoire, this does not necessarily mean it can construct a detector set capable of recognizing ... all strings not in S.*" (D'haeseleer, 1996). He continues by showing that almost all practical matching rules could exhibit these 'holes.' If there are two strings in the *self* set that match each other at k -contiguous bits, for example 00111100 and 11111111 when $k=4$, then any other string that contains these k bits, such as the strings 00111111 and 11111100, can not be detected as nonself because any candidate detector would also match the self strings. D'haeseleer made a *statistical* analysis to estimate the number of strings that would be impossible to detect (D'haeseleer et al., 1996; ?).

Solutions to the Scalability and Coverage Problems In contrast to previous statistical analyses (Percus et al., 1993; D'haeseleer, 1996), Wierzchoń presented a *deterministic* analysis of negative selection (Wierzchoń, 2000; Wierzchoń, 1999). The ability to calculate the exact number of detectable strings was used to construct minimal size detector sets, and a similar analysis allows the number of holes to be computed, i.e. the strings that can not be detected by a given detector set, (D'haeseleer et al., 1996).

Wierzchoń's ability to determine the exact number came from analysis of substring *templates*. A template is a substring of length k of a string of length l , with the remaining $l - k$ bits set to a 'don't care' character such as '*'. For example, the string 100110 has the following contiguous templates of length 4: 1001**, *0011* and **0110. Templates were originally introduced by (Helman and Forrest, 1994). Wierzchoń observed that substring templates can be organized to form a binary tree, and since two templates can overlap there will be a smaller overall number of string holes than might be expected. Analysis of these template trees, allows the construction of much smaller detector sets, thus attacking both the coverage and scalability problems. Although it is impossible to avoid holes when using the k -contiguous bits rule, Wierzchoń showed how to minimize them.

At about the same time as Wierzchoń's work, Hofmeyr and Forrest developed an alternative approach to the coverage problem (Hofmeyr and Forrest, 2000). Figure 1 shows the detectors as the same size and shape, and consequently they can not 'fit' into the hole. However, by using a *permutation mask* is it possible the change their shape

in the state space (for example, by making them thin ovals), thereby allowing holes to be covered by detectors without also covering self strings. The permutation mask is used to re-order a string. For example, applying permutation mask 5-3-1-2-4 to a string '00101' gives the string '11000'. This is, in effect, a way of relaxing the 'contiguous' constraint. By using several permutation masks, which are randomly generated, it is possible to minimize the chance of failing to cover a hole with a detector set.

Another approach (Ayara et al., 2002), based on work by Timmis and de Castro, investigates simple detector generation by random means, with the addition of *clonal selection* (see Section 5). As in Forrest et al's original algorithm, detectors are chosen randomly, but then clonal selection is applied to any detectors that match self strings. The aim is to gradually adjust the detectors until they (just about) no longer match any of the self strings, without the complexity of the deterministic work of Wierzchoń, or Hofmeyr and Forrest's variable shape detectors, and in this respect it was successful. Parthasarathy has reported similar work (Parthasarathy, 2003), as have Kim and Bentley (Kim and Bentley, 2001), who have also explore the benefits of negative selection, clonal selection and methods of self-string generation.

The State-of-the-Art González and Dasgupta described a similar approach to Hofmeyr and Forrest's (Hofmeyr and Forrest, 2000), which uses random sized hypercubes as detectors (González and Dasgupta, 2002). They report that this gives improvements over the traditional form of detector, but do not appear to make comparisons with existing results until later work, which extends the model to real-valued 'negative selection' (Gonzalez and Dasgupta, 2003). They explain that, "*This method does not use positive or negative detection. Rather it tries to find a boundary between normal and abnormal classes*", and their work combines AIS with a standard self-organizing map classification algorithm. In another combination of AIS with existing methods, Gomez, González and Dasgupta present a fuzzy version of negative detection (Gomez et al., 2003).

Recently Esponda, Forrest, Helman and Balthrop have introduced the terms *r-chunks* (Balthrop et al., 2002) and *crossover closure* (Esponda and Forrest, 2003) to negative selection. The *r-chunks* approach simply allows detectors of any size to be used, as opposed to the uniform-sized *k*-contiguous bit detectors of earlier work. This has a surprisingly large improvement on the ability to fully cover the self state space without holes. Crossover closure can be explained by the following example from Esponda's paper (Esponda and Forrest, 2003). Consider two self strings, 0000 and 1011. These may be stored as relations with an attribute for each bit of the strings, where each attribute is either a one or a zero,

R(A1, A2, A3, A4)			
0	0	0	0
1	0	1	1

However, for efficiency reasons, larger tables of this sort of relation scheme might be decomposed into three separate tables, for example three tables of two attributes each (assuming $k=2$),

R(A1, A2)		R(A2, A3)		R(A3, A4)	
0	0	0	0	0	0
1	0	0	1	1	1

To reconstruct the original strings one computes the natural join of the three relations above. However, this will produce more than the original strings; it will return

the *crossover closure* of those strings—i.e. all possible crossovers (single point crossover, in GEC terms) of the original strings.

R(A1, A2, A3, A4)	A1	A2	A3	A4
0	0	0	0	0
0	0	1	1	0
1	0	0	0	0
1	0	1	1	1

The table above shows the total set of strings that are undetectable, X , given the k -width r -chunks in the preceding table. Esponda et al organize their r -chunks in a graph structure (Esponda and Forrest, 2003) to show the relationships between the r -chunks of the members of X . For more details, as well as coverage of positive selection and an up-to-date, comprehensive review of negative selection work at Santa Fe, see (Esponda et al., 2003).

Crossover closure can be reduced to a form similar to Wierzchoń's binary tree template analysis. The main difference is that Wierzchoń worked with k -contiguous bit detectors, where Esponda et al have variable-length r -chunk detectors, and Esponda et al's work appears to have been developed entirely separately from Wierzchoń's work, being inspired by observations of efficient ways of storing relational data in a database.

Ceong et al have developed a system that combines positive and negative selection so that they effectively cross-check the results of each other, and allow for changes in self, over time (Ceong et al., 2003). This is a simple but powerful development.

Bersini and others have taken a different view of the self-nonsel issue, and have introduced the concept of self-assertion in its place (Bersini, 2002). This suggests that the immune system generates its own sense of self, based on its interaction with the world (and the antigenic material therein), by somewhat linking the ideas of self-nonsel and immune networks.

Finally, Esponda and Forrest have developed a prototype 'negative database' based on the principles of negative selection. Unlike a normal database, a negative database stores information about the inverse of the data we wish to store. Since all data can be expressed as a set of strings, and since there are a finite set of strings of any given length, it is possible to generate generalist strings that cover the non-data space using negative selection. This may seem to be a rather odd way of proceeding but this approach has one distinct advantage: it is an extremely secure way of (not) storing data since it can be shown to be an NP-complete problem to decipher the self data, given the nonself data stored in the database (Esponda and Forrest, 2004).

3.3 Typical Applications

Neither this section, nor later, similar sections will attempt to provide an in-depth review of applications because the main focus here is the methods themselves, not their applications. A fairly recent, comprehensive review of applications in all three AIS methods has been made by (de Castro and Von Zuben, 2000). Typical applications are presented here only to give a sense of how each type of AIS may be used for negative detection; these include:

- Change detection (Forrest et al., 1994), in which a negative detection AIS has been used to detect changes made to computer machine code programs caused by computer viruses, to distributed change detection (Forrest and Hofmeyr, 2000), and to network security (Hofmeyr and Forrest, 2000).

- Dasgupta and Forrest's comparative application of negative selection to 'tool breakage detection' in a milling operation (Dasgupta and Forrest, 1995), and their work in benchmark problems (Dasgupta and Forrest, 1999) indicates that it has significant advantages over ANN and some standard fault detection methods.
- Fault detection and diagnosis, such as work by (Ishida, 1993), (Ishiguru et al., 1994) and Bradley and Tyrell's application to build hardware fault-tolerant systems (Bradley and Tyrell, 2002).
- Deaton et al's work (Deaton et al., 1997) that proposed a means of implementing the negative selection algorithm in a DNA computer.
- Network intrusion detection (Anchor et al., 2002), which also combined negative selection with multiobjective evolutionary programming.

Some have questioned the logic of attempting to randomly cover the potentially infinite nonself hyperspace with finite-sized detectors. Ebner et al have suggested that it is instead preferable to generate detectors to cover the *self* strings (i.e. a form of positive detection), because it is usually finite and smaller than the nonself space (Ebner et al., 2002). Moreover, since methods like ANN and support vector machines (SVM) can form a hypersurface that can divide a state space into two (or more) subspaces, what is the point of randomly generating multiple detectors that are likely to leave gaps or contain redundancies and therefore provide less accurate classification? The responses are two-fold (Dasgupta and Forrest, 1999): (i) having multiple detectors allows a non-self event to be approximately located in the state space, and (ii) detectors can be created without requiring any knowledge of the state space, unlike standard techniques. Dasgupta and Nino had compared positive versus negative selection two years earlier (Dasgupta and Nino, 2000). Furthermore, Esponda et al evaluate the benefits and tradeoffs of positive and negative selection (Esponda et al., 2003) and, although they do not specifically reference Ebner et al (Ebner et al., 2002), they present an effective response to their comments. In some cases it may be more appropriate to cover the self strings with detectors; in other cases negative selection is appropriate. So, is negative selection distinct and effective?

3.4 Applying the Usefulness Criteria

Consider again the distinctiveness criteria of Section 2. What are the symbols, expressions and processes in negative selection?

D.1-Symbol Distinctiveness: The symbols are binary (or real-valued) numbers. These symbols are frequently used in many methods and are not distinct. [*D.1 = No*]

D.2-Expression Distinctiveness: These symbols are organized in vector expressions that are called detectors. Very many other methods use vectors of values: most GEC systems and some ANN systems, to name just two paradigms. However, detectors can be used differently to the vectors in other systems, e.g. they can be distributed easily across many computers, and this provides some degree of differentiation.

One might argue that the meaning of the expressions in positive selection is different to the meaning of those expressions in negative selection, because they express different aspects of the detection problem, but the counter argument is that what is really being defined (in both positive and negative selection) is the dividing line

between the positive and negative cases. If this is true, then the expressions represent the same entity, which is the dividing line. Others have specifically focused on exploring the formal nature of positive and negative detection in detail (Esponda et al., 2004). [*D.2 = No, with some exceptions*]

D.3–Process Distinctiveness: The central question is, are the processes on these vector expressions reducible to existing processes? There does not seem to be any similarity to ANN methods (although others may wish to attempt a proof of such similarity!), but there is a link with at least one GEC-based method.

Hofmeyr and Forrest make a comparison between negative selection and classifier systems (Hofmeyr and Forrest, 2000). They then set out a table of synonymous terms between classifier systems and negative selection, and make a connection in all but one case (there are multiple representations in negative selection). However, Hofmeyr and Forrest are not saying that classifier systems are indistinct from negative selection, “*There is no direct analog of the IS negative-selection algorithm in classifier systems except for the learning rules ... under which new classifiers are generated.*,” which indicates a degree of distinctiveness. They highlight other distinct properties, e.g. the ‘condition’ in a classifier system is in the alphabet 0,1,# not the k -contiguous bits of negative selection (Hofmeyr and Forrest, 2000). Similar work is presented in (Hofmeyr and Forrest, 1999) where they say, “... [*although*] the AIS outlined ... resembles the architecture of a classifier ... most of the details are different ... [*and*] ... there is no direct analog of our negative selection algorithm in classifier systems.” In other words, they consider the overall design of negative selection to be distinct from classifier systems.

Dasgupta made a comparison between AIS and ANNs (Dasgupta, 1997) but much of the discussion simply places the symbols, expressions and processes of both systems alongside each other, without establishing any form of equivalence. He did, however, note a series of differences (and some similarities) in the processes of both methods. Overall, however negative selection appears distinctive.

Observe that negative selection is a form of positive-only learning, using ‘positive-only’ in the sense used by inductive logic programming (ILP) (Muggleton, 1996). In both negative selection and some forms of ILP, only positive examples are available (i.e. examples of self, or some other class); there is no direct knowledge about the nature of nonself, just the assumption that nonself is equivalent to `not (self)` and therefore defined exclusively by reference to self. There may be potential in sharing ideas from both fields but, this does not appear to have been reported yet, with the possible exception of (Nikolaev et al., 1999). The question is, are the two learning processes the same? There certainly are some similarities in their symbols and expressions. As Esponda et al have pointed out (Esponda et al., 2003) instances of self can be converted to relational format, and it is exactly this format that is used in ILP. The organization of relations into a tree or lattice, as suggested by Wierzchoń and Esponda et al, is also common in ILP. However, the processes on this lattice seem to differ: in ILP, search is the most common operator; in negative selection the lattice is used to count the number of strings. Negative selection appears to be sufficiently distinct here too. [*D.3 = Yes*]

The effectiveness criteria of Section 2 require that a method is either alone, better or faster in its ability to perform a task, and that the task should be of some value. Is this true of negative selection?

E.1—Does the Method Provide Unique Results? D’haeseleer, writing in 1996 before some of the recent advances in efficiency, described above, acknowledges that negative selection may not be as effective as some other systems for detecting change but, he says, the security of such specific systems are liable to be circumvented, and negative provides a generalized ‘safety net’ that could detect change events (D’haeseleer, 1996). Other advantages he lists include the ability to check parts of objects for change (rather than whole objects); several different detector sets can be applied to the same object, making detection failure exponentially less likely with the number of detector sets used. Related to this, subsets of the detector set can be used in an autonomous, distributed manner (D’haeseleer, 1996). The negative database approach also seems to be entirely unique. [E.1 = Yes]

E.2—Does the Method Produce Better Quality Results than Existing Methods?

Dasgupta and Forrest list a number of other techniques that have been used for anomaly detection, namely, control charts, model-based methods, knowledge-based expert systems, pattern recognition and clustering, hidden Markov models, and neural networks (Dasgupta and Forrest, 1999). Other methods exist, such as Parzen windows (Lane and Brodley, 1997). Dasgupta and Forrest have also explored the effectiveness of negative selection relative to an ART ANN (Dasgupta and Forrest, 1999; Dasgupta, 1997), finding the negative selection AIS was more precise in detecting changes in a signal than the ART ANN. [E.2 = *In some cases, perhaps*]

E.3—Is the Method Faster than Existing Methods? There does not appear to have been any investigation into the speed of negative detection, relative to other methods. This may be because time complexity has been a major problem for negative detection, although this appears to be changing (Esponda et al., 2003). [E.3 = *No*]

Dasgupta appears to be alone in comparing negative selection to other methods (Dasgupta and Forrest, 1999; Dasgupta, 1997), in this case to ANNs, and this limits the confidence with which one can declare negative detection to be effective. In part, this is because AIS is still a young field, in future, however we should expect more comparisons with the performance of other methods on the same application.

Another issue, which impacts on the potential usefulness of this form of AIS, is how appropriate negative (and positive) selection are for intrusion detection. By their very nature, computer intrusions are planned and initiated by intelligent agents (usually humans) and this contrasts with the highly-evolved-but-dumb behavior of bacteria and viruses. One argument might be: “If the natural immune system were under attack by intelligent agents it is likely that it could be made to fail, and indeed this is exactly what happens during treatments given after skin grafts and transplants.” However, others, such as Stephanie Forrest, powerfully argue that, “... in the biological warfare arena, the best that engineers have accomplished so far (thankfully) is minor tweaking of pathogens that were evolved by nature,” (personal correspondence) and these pathogens, such as the Ebola virus, are worse threats to human health than anything that man has engineered. In the same way, negative selection can be a very strong approach for intrusion detection.

In any case, there certainly are some classes of applications in which negative selection is the method of choice. For example, consider the abstract (but useful) case where the self strings themselves are not directly available, only a function that determines whether a string belongs to the self set or not. In this case, negative selection

would probably be the most effective algorithm available, since the alternate methods listed above would fail.

4 Danger Theory: An Alternative Approach?

This section considers an emerging AIS technique for intrusion detection that is based on an abstraction of Matzinger's *danger theory* (Matzinger, 2002). Danger theory AIS is given a section to itself because it is a distinct, fast growing alternative/addition to negative detection.

4.1 Background Immunology

By the mid-1990s, immunology had made several modifications to the self-nonsel self theory. The original self-nonsel self theory did not fit experimental observations, such as why there is no immune response to bacteria in our gut, or to air, both of which are clearly nonself, and Matzinger suggested a radical, alternative approach. What if, instead of responding directly to a nonself element of some sort, the immune system responds to cells that are under stress; cells that are raising an alarm to some form of danger, such as an attack from a bacterium or virus? Matzinger characterizes this danger theory as a means of detecting 'some self from some nonself,' thus explaining why there is no immune response to harmless nonself, but why there is usually an immune response to harmful self, such as cancerous cells. This is the basis of danger theory.

Cells can die in two ways: via *apoptotic*, normal death that has been requested by the body's internal signaling system, or via *necrosis*, a form of unexpected death caused by something going wrong with the cell, which often causes an inflammatory response. Matzinger suggested that the immune system is particularly activated by cell necrosis. Thus, under the danger theory, the immune response is contextualized to the location in which necrosis is occurring and is no longer a system-wide response. Despite Matzinger's departure from more established theories, such as the work of Janeway, e.g. (Medzhitov and Janeway, 2002), both require *two* signals for an immune response to be initiated. This helps to avoid false positive reactions in nature (causing autoimmune effects), and may be of use in AIS. However, the major problem with the Matzinger's theory is that the exact nature of the danger signal(s) is still unclear.

4.2 The Development of Danger Theory AIS

It appears that danger theory might help intrusion detection systems by focusing attention only on those internal or external events that are likely to be harmful—generally a smaller subset of events than the nonself subset. Whether or not this danger theory has validity in any natural immune system response is of little importance here because danger theory AIS (DT-AIS) is an abstraction of this process for purely computational purposes; it is not a model of the immune system. Several researchers are pursuing this approach (Aickelin and Cayzer, 2002; Williamson, 2002; Burgess, 1998).

It may seem pointless for DT-AIS to simply change the self-nonsel self dichotomy for a danger-nondanger dichotomy, but there are some important differences. The concepts of self and nonself are both relative to the self strings, which are not necessarily the full set of true self, may change over time, and may contain many features or attributes. However, the concepts of 'dangerous' and 'non-dangerous' are *grounded* in undesirable events, and a detection system based on these concepts should, in principle, only report the specific attributes or features that are causing concern. These danger signals may be positive—e.g. the presence of an event or state such as high memory or disk activity, frequency of file changes, unusual signals (e.g. SIGABRT in UNIX), and the presence

of nonself (Aickelin and Cayzer, 2002)—or the signals may be negative, such as the absence of such signals or states. In addition, the possibility of requiring two signals to be present before an immune response is initiated seems to be a way of minimizing false positives. (Secker et al., 2003) describe one possible application, email mining and spam detection. Their paper includes some pseudocode, but as yet the system has not been fully implemented, mainly because the nature of the danger signals is unclear.

4.3 Assessment of Its Promise

In the natural immune system, it is not immediately clear which signals are danger signals (if any), and it already appears that this is the major problem in implementing a DT-AIS too. Another drawback is that the DT-AIS system will apparently have to wait for damage to occur to self at least once before any protection steps can occur, because it requires examples of dangerous states. This is not the case with negative detection, which can give a response based solely on positive examples data of self.

Issues of scaling also need to be considered. DT-AIS is a more complex theory than standard negative selection (e.g. the algorithm in (Secker et al., 2003)) because it is based on a much expanded version of Burnet's original work in negative selection and the proliferation of B-cells (Burnet, 1959). It is also unclear how best to involve a human in the loop (if at all). Nevertheless, the possibility of using DT-AIS to obtain key features from a set of attributes is an interesting problem, with relevance in many areas. If DT-AIS can find an automated solution to this problem it would have great benefits within, and beyond, AIS.

There is very little point in applying the usefulness criteria (Section 2) to such a new development in AIS, but current developments are distinct in their use of existing novel AIS processes (Secker et al., 2003), and (Aickelin et al., 2003) have shown that DT-AIS might favorably compare with existing (non-AIS) approaches to intrusion detection.

5 Artificial Clonal Selection and Hypermutation

5.1 Background Immunology

Whereas negative selection and danger theory center on the *detection* of nonself or danger signals, in clonal selection the focus is on how B-cells (and T-cells) can *adapt* to match and kill the invader. The immune system's ability to adapt its B-cells to new types of antigen is powered by processes known as *clonal selection* and *affinity maturation by hypermutation*. In AIS literature, the shorthand 'clonal selection' is often used to refer to both processes.

According to Burnet, biological clonal selection (Burnet, 1959) occurs to the degree that a B-cell matches an antigen. A strong match causes a B-cell to be cloned many times, and a weak match results in little cloning. These 'clones' are mutated from the original B-cell at a rate inversely proportional to the match strength: a strongly matching B-cell mutates little and thus retains its match to the antigen to a slightly greater or lesser extent; a weakly matching cell mutates much more. See (French et al., 1989) for a more recent appraisal of the role of somatic hypermutation.

As mentioned above, since 1959 there have been improvements to Burnet's theory, with respect to the way antigens are recognized, whether as nonself or as dangerous, but his basic principles of clonal selection and affinity maturation by hypermutation are sufficient for the purposes of AIS clonal selection.

5.2 The Development of Artificial Clonal Selection and Networked Artificial Selection

Early Work in Artificial Clonal Selection Early work in artificial clonal selection grew out of computational models of the processes of clonal selection, such as Weinand's work (Weinand, 1990), and more recently it has become an abstract method for optimization, search and pattern recognition. There have been several degrees of emphasis between these two poles, such as Forrest and Smith's more abstract work, and Chowdhury's more immunologically emphasized analysis of the computational properties of the immune system (including immune network elements, as well as clonal selection) (Chowdhury et al., 1990; Chowdhury and Stauffer, 1992) that suggested that clonal selection might be used for computational intelligence. Hightower, Forrest and Perelson considered clonal selection from the point of view of the Baldwin Effect (Hightower et al., 1996) and pattern matching (Forrest et al., 1993).

It was Fukuda, Mori and Tsukiyama's often overlooked work (in the West at least) that first developed an algorithm that included an abstraction of clonal selection to solve computational problems (Mori et al., 1993; Fukuda et al., 1998). They applied their algorithm to optimization applications, both uni-modal and multi-modal, to scheduling and resource-allocation problems, and report its use in a semi-conductor production line (see (Dasgupta and Yu, 2003) for complete references). Their algorithm is quite complex—relative to later clonal selection algorithms—and contains a direct means of concentration control that uses information theory to help choose the next set of antibodies. Since their work also has a form of network control, it will be discussed in detail in Section 6; nevertheless, it was the first to explicitly use an abstraction of clonal selection and, although this approach has been largely ignored, their use of information theory is now being revived, with a new treatment, and applied to clonal selection, and negative selection (Nicosia et al., 2003).

De Castro and Von Zuben's CLONALG The artificial form of clonal selection has been popularized mainly by de Castro and Von Zuben, beginning with an algorithm they called CSA (de Castro and Von Zuben, 1999), which was then modified and renamed CLONALG (de Castro and Von Zuben, 2000). CLONALG currently exists in two forms (de Castro and Von Zuben, 2002)—one for optimization tasks and one for pattern matching, however the pattern matching algorithm has only recently been more fully investigated (White and Garrett, 2003). CLONALG has the advantage of being simple to implement, relative to previous work such as Fukuda's.

CLONALG for Optimization: The basic CLONALG optimization algorithm is as follows (see (de Castro and Von Zuben, 2000) for details):

1. *Initialization:* Create an initial, random population of antibodies, P_0 . Iterate steps 2–7 if a predefined termination condition is not met.
2. *Evaluation and Selection 1:* Select a subset, F , of the fittest antibodies from P_t according to some fitness function, $f(ab_i)$.
3. *Cloning:* For each $ab \in F$, create a set of clones, C_i , such that $|C_i| = nc(ab_i)$. The set of all clones, $C = \bigcup_i C_i$.
4. *Mutation:* Mutate each clone $c \in C$ by a function $am(f(ab_i), \rho)$ Add the mutated clones, C' , to P_t to give P'_t .
5. *Evaluation and Selection 2:* Select a subset, F' of the fittest antibodies from P'_t .

6. *Diversity*: Add r randomly generated new B-cells to F' give a new population P_t'' .
7. *Death*: Retain only the best $|P_t|$ members of P_t'' to give P_{t+1} ; all other B-cells are considered to have died.

The effect, on average, is that each successive generation of B-cells tends to be closer to the state space optima than the previous generation. The addition of random members at each generation provides diversity (to search unexplored areas of the state space) for still better approximations to the optima, but the evolutionary pressure lessens as all antibodies tend to their (local) optima. De Castro presents results that show convergence can be on several optima at the same time (multi-modal optimization) (de Castro and Von Zuben, 2000).

CLONALG for Pattern Matching: The version of CLONALG for pattern recognition is similar to the optimization version. However, when training the system on a set of patterns, a step is added that selects the single best member of the population for each generation. This population member is called a *memory cell*, since it remembers the best approximation to the pattern being shown to the system, and it is retained if it has a higher fitness than any previously stored memory cell for the pattern currently being shown to the system. One memory cell is allocated for each pattern to be remembered (de Castro and Von Zuben, 2002), giving a set of memory cells, M .

When training is complete, a pattern p is classified by the class of the memory cell that has the minimum distance from p . I.e. $p^{class} = m_i^{class} \wedge \min(dist(p, m_i) : \forall m_i \in M)$, where $dist(x, y)$ is an appropriate distance function between the pattern and winning memory cell. De Castro and Von Zuben only provided a proof of principle (de Castro and Von Zuben, 2002) that the pattern matching form of CLONALG could work, but it was enough to show it had promise as a pattern recognition method. In fact, this was not the first paper on this subject: Forrest et al. had studied the use of genetic algorithms (GAs) for pattern matching in the context of the immune system (Forrest et al., 1993). Their work used the binary schema approach common in GA work, and they provided a formal analysis of the pattern matching abilities that draws heavily on the analyses Forrest and others made on GA effectiveness at certain types of problem.

White and Garrett have recently presented a study that applied CLONALG to pattern matching. They also introduced a more efficient version of the pattern matching form of CLONALG called CLONCLAS, and compared both CLONALG and CLONCLAS to simple Hamming distance methods in order to explore whether the complexity of clonal selection was necessary for pattern matching (White and Garrett, 2003). Their results show that clonal selection can be significantly more effective than Hamming distance methods (particularly CLONCLAS), although with increased computational cost.

One of the main problems with both forms of CLONALG is its scalability. Since the number of clones rises in proportion to the value of $f(ab)$, the number of objective function evaluations will quickly increase as the majority of the population approaches the (local and global) optima. If the $nc(x)$ function is carefully formed then the average number of clones per generation may asymptotically approach a constant value, otherwise it will be unbounded. In both cases, however, it can cause the algorithm to run very slowly.

The State-of-the-Art The method(s) of Nicosia and Cutello are similar to CLONALG (e.g. (Cutello and Nicosia, 2002)) in its general form but have some important additions. Perhaps the most important of these is the introduction of a probabilistic half-life for each B-cell, and the re-introduction of information theory into clonal selection

(Nicosia et al., 2003; Cutello and Nicosia, 2002). Fukuda et al had used information theory in their approach (Fukuda et al., 1998), but Nicosia and Cutello have simplified its application to clonal selection and used it in a novel terminating condition: if the information measure drops to zero then the run stops. As well as this, they define the initial population of vectors by a Gaussian distribution, as opposed to the more common uniform distribution. There does not appear to be one single algorithm, like CLONALG, produced at Catania, rather they improvise on a theme as required by an application or theoretical exploration (Nicosia et al., 2003; Cutello and Nicosia, 2002; Castiglione et al., 2001).

In contrast, AIRS is a form of clonal selection algorithm that has been developed from the AINE immune network (pronounced 'Ann-ee'), see Section 6.2. AIRS still uses many of the elements from its immune network heritage, such as an unusual form of resource limitation to control the population size, and the concept of the Antigen Recognition Ball (ARB). An ARB is a data structure that represents multiple, identical antibodies, thus a population of ARBs can represent a much larger population of antibodies in an efficient manner. The ARB also introduces the concept of cell concentrations into clonal selection, something that is totally absent in de Castro and Von Zuben's work, for example. AIRS is commonly used for supervised classification, which has obvious connections with pattern matching, and has been comprehensively compared to the LVQ method (Kohonen, 1990) with mixed results (Goodman et al., 2002; Watkins and Timmis, 2002). Although AIRS can outperform LVQ, it tends to do so by requiring more computing resources.

De Castro's and Von Zuben have largely moved away from pure clonal selection, towards a method, known as AINET, that combines CLONALG with aspects of immune network models (de Castro and Von Zuben, 2001). The resulting network has been shown to give better results than CLONALG in every test that they have performed. Garrett has argued that clonal selection and immune networks are strongly related, to the point that clonal selection may be considered a simplification of some forms of immune network (Garrett, 2003). If this is the case, then the question, 'how much network is best?' may be more appropriate than, 'is clonal selection better than immune networks?'

Garrett has recently presented an attempt to remove all the parameters from a clonal selection method, or rather to remove all parameters that require tuning (Garrett, 2004). This approach attempts to predict the final fitness of various internal parameters, and self-evolves various parameters during a single run. As a result, the method can be used without any prior testing of parameter values on the given objective function; this is useful when time is limited, or must be as accurate as possible.

5.3 Typical Applications

Typical applications of clonal selection include the following:

- CLONALG has been applied to uni-modal, combinatorial and multi-modal optimization, and to the problem of initializing the centers of radial basis functions (de Castro and Von Zuben, 2001).
- The use of CLONALG has been suggested for various types of pattern recognition (de Castro and Von Zuben, 2002) and shown to work by (White and Garrett, 2003).
- Cutello and Nicosia have applied their clonal selection method to the graph coloring problem, to two NP-complete problems, and multiple character recognition problem (Cutello and Nicosia, 2002; Nicosia et al., 2003).

- (Hart et al., 1998) used a clonal selection-based method in automated scheduling.
- Greensmith and Cayzer have recently suggested the use of AIRS in document classification (Greensmith and Cayzer, 2003).

5.4 Applying the Usefulness Criteria

Several commentators have informally suggested that clonal selection is just another form of GEC algorithm. To a degree this is true, because one of the central purposes of natural clonal selection and affinity maturation is to ‘out-evolve’ bacteria, viruses and other invaders, but is artificial clonal selection distinct from GEC and other methods?

Wierzchoń makes a distinction between GA-based methods, used in function optimization, and approaches that are inspired by the processes of the immune system (Wierzchoń, 2002), where he also introduces a method similar to CLONALG and presents a helpful list of GA and AIS methods used for optimization. However, as discussed in Section 2, appealing to the source of inspiration appears to have limited value in distinguishing between methods.

D.1 - Symbol Distinctiveness: Some types of artificial clonal selection use binary vectors, and some use real-valued vectors, but in both cases there is nothing particularly distinct about the symbols used. [*D.1 = No*]

D.2 - Expression Distinctiveness: The organization of these symbols is also not distinct from very many other methods, the only notable exception is perhaps the ARB, used in Watkin and Timmis’ work (Watkins and Timmis, 2002). [*D.2 = Some distinctiveness*]

D.3 - Process Distinctiveness: One of the main questions asked about clonal selection is, “How does clonal selection differ from other GEC algorithms, such as evolutionary strategies?” Evolutionary strategies (Bäck and Schwefel, 1993)² control the degree of mutation of real-valued population members by one or more adjustable *strategy parameters*. These parameters are altered each generation, either being multiplied or divided by a constant value, usually 1.3 (Rechenberg, 1994). Mutation-based genetic algorithms, which use mutation to find solutions, also exist (Lau and Tsang, 1996).

Where clonal selection differs from other GEC methods is in its mechanism for adjusting mutation. Unlike ES and mutation-based GAs, which have constants that control the degree of mutation (or the degree of change in mutation), clonal selection’s degree of mutation is *functionally* related to the fitness of the solution—solutions with small deviation from the optimum have small degrees of mutation. It should be pointed out that Steve Hofmeyr did do some work on this in his MSc thesis (unpublished), relating it to GAs, although the method for adjusting mutation rates was different. This was the only work that could be found that was similar to the AIS approach of adjusting mutation rates and the number of clones.

This is more than semantic gymnastics. An ES *evolves* low mutation rates for its high fitness members; clonal selection *functionally defines it*. Although the results are similar, the mechanisms are different, and this difference can be measured. Walker and Garrett compared the performance of ES and (real-valued) clonal selection over a number of functions, with different dimensionalities and found that

²Earlier versions exist in German, as cited in Back and Schwefel’s paper

clonal selection tends to perform better than ES at low dimensionalities, whereas ES performs better at higher dimensionalities (Walker and Garrett, 2003). These results were shown to be true for stationary functions and for dynamic functions, in which the locations of the optima change over time.

Similarly, the functional relationship between a vector's fitness and the number of clones it produces is unusual, if not unique to clonal selection. Not all clonal selection algorithms have this feature: in the optimization version of CLONALG, for example, the number of clones value is held constant! For clonal selection in general, however, there is a functional relationship between the fitness of a population member and the number of clones it produces. Garrett has investigated clonal selection algorithms that evolve the function that decides this value (Garrett, 2004) and compared them to an ES that has been pre-parameterized in the number of children (clones) that each parent produces. The results indicate that parameter-free clonal selection algorithms are more effective than pre-parameterized ES, over the four functions used for testing.

This is not to say that hypermutation is a new idea in GEC research. Cobb and Grefenstette have long been interested in its effects (Cobb, 1990; ?) but, in those cases, the degree of hypermutation was linked to the diversity of the population and not the fitness of a given solution vector. Although it may be possible to find GEC algorithms that employ one or other of the distinctive elements of clonal selection, only clonal selection appears to contain both of them in a single algorithm, and only clonal selection has such a clearly functional relationship between the fitness of a solution and its amount of mutation and number of clones. [D.3 = Yes]

E.1 - Does the Method Provide Unique Results?: There do not appear to be any unique abilities of clonal selection algorithms. [E.1 = No]

E.2 - Does the Method Produce Better Quality Results than Existing Methods?:

(Hart et al., 1998) have shown that scheduling is much more robust using clonal selection than using a simple GA. However, even if one can show that clonal selection is the best choice of algorithm for some valuable subset of objective functions, de Castro's recent research has shown that networked versions of clonal selection are usually more effective (de Castro and Von Zuben, 2001).

As well as noting that clonal selection was better than ES at low dimensions (by several orders of magnitude), Walker and Garrett (Walker and Garrett, 2003), also noted that at all dimensionalities the graph of ES optimization tended to be asymptotic in style (given a log scale on the fitness axis), whereas the AIS graph continued to optimize at roughly the same, slower, rate. This may suggest that clonal selection has more consistent evolutionary pressure than the ES, and would eventually out-perform an ES, given enough time; however, their work did not properly address this point. A similar analysis has been reported by Gaspar and Collard, although their work modified a GA to give an immune network, as opposed to a clonal selection algorithm, and the results were not as detailed (Gaspar and Collard, 1999). [E.2 = *Sometimes, but clonal selection may be superceded by aiNet-like algorithms*]

E.3 - Is the Method Faster than Existing Methods?: CLONALG can be very slow to run because the number of clones, and therefore the number of fitness evaluations, increases with the increase in the average fitness of the population; this is because

the number of clones is functionally related to the fitness of the population. AIRS is often more resource-hungry than Kohonen's LVQ (Goodman et al., 2002) but performs very well. New methods are appearing that are more efficient, but they have not yet been compared to existing methods in terms of their speed and efficiency. [E.3 = Not yet clear]

Clonal selection is an interesting evolutionary approach that has relevance not just for immune network research, but also for GEC research. The work of Nicosia et al is a promising new, efficient approach to building an abstraction of clonal selection, and the connection with immune networks means that improvements to clonal selection are doubly important.

6 Artificial Immune Network Models

6.1 Background Immunology

So far, we have seen that part of an antibody (known as its *paratope*) will bind to part of an antigen (known as its *epitope*). Immune network models³ (Jerne, 1974) go further, in that antibodies also have epitopes, which can be bound by other antibodies' paratopes. In all cases, the entity presenting the bound epitope is then eliminated or repressed, and the antibody presenting the active paratope is proliferated. Under this assumption, a network of stimulatory and suppressive interactions exists between antibodies that affects the concentrations of each type of antibody, and it has been shown that this might allow for a form of associative memory, somewhat like a Hopfield neural network or sparse distributed memory (Hart and Ross, 2002).

An abstracted mathematical model of a Jerne's immune network theory was first suggested by (Farmer et al., 1986). As well as assuming that Jerne's theory was correct, Farmer et al. consciously made the simplifying assumption of a single epitope-binding region on each antibody and antigen, and a single paratope-binding region on each antibody. Having presented their model, they also discussed the possibilities for using these abstracted immunological descriptions as computational tools for machine learning.

6.2 The Development of Artificial Immune Networks

The algorithms for immune network models are tightly linked to two equations. For clarity in comparing types of network model, only the equations, not the algorithms, are presented here. Algorithmic details may be found in the cited papers.

The 'Farmer-Packard-Perelson' (FPP) Model The model of (Farmer et al., 1986) (FPP) is as follows. Consider a set of antigen and a set of antibody types, where 'antibody type' means a concentration of antibodies that share exactly the same binding features. Both antigen and antibodies are modeled as binary strings, and the m_{ij} matching affinities between the two antibodies are defined as,

$$m_{ij} = \sum_{k=1}^{rng} G \left(\sum_{n=1}^l e_i(n+k) \oplus p_j(n) - s + 1 \right), \quad (4)$$

where ' \oplus ' is the complementary XOR operator; k is the offset, measured in bits, between the paratope and epitope; $e_i(n)$ is the n 'th bit of the epitope; $p_j(n)$ is the n 'th bit

³Strictly speaking, these are not models, they are abstractions of immune processes; however, this is the term generally used.

of the paratope, and s is a threshold since $G(x) = x$ if $x > 0$ and $G(x) = 0$ otherwise. The number of bits in a string, $l = \min(\text{len}(e_i), \text{len}(p_j))$, $\text{rng} = l - s$, $s \leq l$. Therefore, the value of k is in the range $-\text{rng} \leq k \leq \text{rng}$. Note that the variable rng has been deduced here from (Farmer et al., 1986) to clarify k 's range. Also note that Equation 4 is *not* symmetric because m_{ij} means "epitope i binds with paratope j ," and m_{ji} means "epitope j binds with paratope i ."

Equation 4 was then used in a dynamic differential equation for modeling the concentrations of changing antibody types. Given N antibody types, with concentrations $\{x_1, \dots, x_N\}$ and n antigen types, with concentrations $\{y_1, \dots, y_n\}$, the change in concentration of a given x_i was modeled as,

$$\frac{dx_i}{dt} = c \left[\sum_{j=1}^N m_{ji} x_i x_j - k_1 \sum_{j=1}^N m_{ij} x_i x_j + \sum_{j=1}^n m_{ji} x_i y_j \right] - k_2 x_i. \quad (5)$$

The $x_i x_j$ and $x_i y_j$ elements model the probability that a paratope and epitope will be close enough to attempt to bind, since high concentrations of either will increase this probability. The first term in Equation 5 models the stimulation of an antibody type's paratope by the epitope of another antibody type. The second term indicates the repression of an antibody type due to its epitope binding with the paratope of another antibody type. The third term models the stimulation of an antibody type by binding with the epitope of an antigen. The constant k_1 indicates a possible inequality between the stimulation and repression terms. The equation ends with $k_2 x_i$ (where $k_2 > 0$), a 'death term' that removes a quantity of x_i antibodies. Farmer et al advise that the value of k_2 is adjusted until constant population size is obtained, as in the natural immune system. Finally, c is a rate constant, common in kinetic equations, that depends on the number of collisions per unit time and the rate of antibody production stimulated by a collision.

Two negative issues arise from the FPP model. Firstly, it is strongly tied to the use of binary data or, more precisely, to a binary representation of amino acid binding. It is not clear whether it is valid to use the FPP model for other types of data, such as binary representations of base-10 real-valued variables. For example, the binary numbers for 63 and 31 only differ by 63 having an extra '1' bit, and yet these numbers are not clearly related in base-10.

Secondly, standard kinetic equations (Cleland, 1963) suggest that there should almost certainly have been coefficient constants for the first three terms of Equation 5, not just k_1 for the second term, since this allows any term to dominate the others or to become insignificant.

Having described the immune network in these terms, Farmer et al drew comparisons between the immune network and Holland's classifier system. In later work, Farmer broadened the scope of this comparison by suggesting that immune network AIS, classifier systems, autocatalytic reaction networks and artificial neural networks are all instances of connectionist systems (Farmer, 1990).

The Model of Fukuda et al The work of Fukuda (e.g. (Fukuda et al., 1998; Fukuda et al., 1993) has some similarities with the FPP model, discussed above, in two respects: it represents the paratopes as binary vectors, it is a network model, and it is concerned with concentrations of antibody types. However, it differs in many other ways: the equations are not specifically dynamic (no derivatives); there are many more equations, each modeling a part of the immune system, such as the informative entropy of the antibodies, the affinity between two antibodies, the affinity between each antibody and

the set of antigen, the proliferation and suppression of antibodies, and so on. Nevertheless, the idea is the same as in the FPP model: simultaneously maximize the affinity of each antibody to the antigen (so that there will be good solutions) and minimize the concentration of any antibody type (so that the population remains diverse).

Fukuda et al apply their method to functions such as the Shubert function (Fukuda et al., 1998) and to real-world applications. Their results are impressive, and they demonstrate that it is possible to do multimodal optimization over the memory cells, while doing global optimization with the population of antibodies. See (Fukuda et al., 1998) for more information.

The ‘Timmis-Neal’ Model Hunt and Cooke took a similar approach to Farmer et al (Cooke and Hunt, 1995), with some modifications. In the FPP model, it was assumed that *all* antigen and antibodies will interact probabilistically; in Hunt and Cooke’s model, the effects of an antigen were localized to k -nearest neighbors of a random point in the network. They also took a slightly different approach to the process of creating and destroying B-cells.

Timmis and Neal’s work (Timmis et al., 2000) originally extended Hunt and Cooke’s work; however, after several implementational problems they decided to significantly, and fundamentally, simplify the FPP model. Their model was a further abstraction of Equation 5, removing the ‘death term’ and constants, and changing the representation of the data from binary to real-valued numbers. This change necessitated a *similarity* measure in place of the *complementary* measure of the FPP model, and it also required the data to be normalized. Furthermore, the change in representation prevented variable-length strings being used as paratopes and epitopes. Finally, they dispensed with concentration measures for the antibodies and antigen.

This ‘Timmis-Neal’ model introduced the idea of a network affinity threshold (NAT), which is used to limit the connections between two antibodies to those that are the strongest, to control the density of the network. Their algorithms have varied over time, but the similarity measure and network equations are generic. Their equations can be re-written in a similar form to Farmer et al’s, for comparison. First the distance metric is,

$$m(x, y) = H \left(\sqrt{\sum_{n=1}^l (x(n) - y(n))^2} \right), \quad (6)$$

i.e. N -dimensional Euclidean distance, such that,

$$\begin{aligned} m(x, y) &= H(x) && \text{if } H(x) \leq NAT \\ m(x, y) &= 1 && \text{otherwise.} \end{aligned}$$

The results of $m(x, y)$ are normalized to fall on or within the unit sphere, i.e. $0 \leq m(x, y) \leq 1$. The dynamics of the Timmis-Neal model are defined by,

$$\begin{aligned} sl &= 1 + \sum_{j=1}^N (1 - m(x_i, x_j)) - \sum_{j=1}^N m(x_i, x_j) - \sum_{j=1}^n m(x_i, y_j), \\ &\text{if } sl > \theta \text{ then } x_i = x_i + k(sl) \text{ else } x_i = x_i. \end{aligned} \quad (7)$$

with the NAT scalar selecting only those antibodies that are close enough to each other. The overall limits of sl can be shown to be $(1 - 2N) \leq sl \leq (1 + N)$.

The Timmis-Neal model’s use of real-valued data led to several difficulties. Firstly, it forced the use of a *similarity* measure (Euclidean distance) in place of the FPP model’s

complementary measure, and the effects of such a change were not evaluated. Secondly, the data required normalization, and the effects of normalization were not analyzed. Consider a two-dimensional vector (x_1, x_2) ; if x_1 is in the range $[-100, 100]$ and x_2 is in the range $[-0.01, 0.01]$ then normalizing the data will significantly change their relative sizes. Thirdly, it made it difficult to implement variable-length strings, which limits application to homogeneous data that can match in only one way (Freitas and Timmis, 2003).

In the original Timmis-Neal model, the network grew exponentially due to the lack of a death term. It could therefore only be run for a few evaluations of Equation 7 (Knight and Timmis, 2001), and in any case it suffered from premature convergence. Others have pointed this out too (Nasaroui et al., 2002a). Timmis prevented population explosion by introducing competition for resources, a resource-limitation mechanism that probabilistically chooses the best members of the current population for the next generation, and removes B-cells with no resources. However, since Equation 7 inefficiently models individual cells, as opposed to the FPP model which models concentrations of cells, Timmis also introduced the ‘artificial recognition ball’ (ARB), mentioned above, to represent several identical B-cells in one data structure; this formed a new resource-limited immune network model known as RLAIS (Timmis and Neal, 2000), and later AINE. Since these changes, Neal has recently returned (to some degree) to the FPP model by reintroducing a death term, and has successfully produced a self limiting system (Neal, 2003; Neal, 2002).

Other Immune Network Models One area of immune network research that has not yet been mentioned is investigation of the dynamics of such networks. In particular, the connection between immune networks and chaos is important. De Boer, Kevrekidis and Perelson specifically investigated the relationship between chaos and oscillating states in immune networks (De Boer et al., 1992), and Bersini and Calenbuhr report experiments with frustrated chaos in small immune networks (Bersini and Calenbuhr, 1997). Both of these reports illustrate the much-overlooked fact that immune networks have to be highly abstracted before they become tame enough to be used in clustering or optimization. Garrett has discussed several of these simplifying assumptions, such as (i) the use of only a single binary or real-valued vector to represent both the epitope and paratope of an antibody; (ii) the false assumption that antibodies will bind best when similar, and (iii) generally simplifying both an antibody and antigen to have only one binding element. In nature, antibodies have at least two paratopes, and antigens may have several epitopes (Garrett, 2003).

Others have investigated alternatives to the FPP-style ordinary differential equation (ODE) models of immune networks. Stadler, Schuster and Perelson modeled the immune network using replicator equations (Stadler et al., 1994), i.e. equations that use Dawkin’s idea of a *replicator*, an abstract entity that gets copied and whose inherited properties affect the copying process (Dawkins, 1976). Replicator equations are differential equations of the form,

$$\frac{dx_k}{dt} = x_k \left[F_k(x) - \sum_{j=1}^n x_j F_j(x) \right], \quad k = 1, \dots, n, \quad (8)$$

and such equations can be used to model a large variety of natural systems, including immune networks.

Ishida also notes that models of immune networks tend to be based around differential equations of one sort or another. He demonstrated that immune networks can,

instead, be modeled by finite state automata (particularly state propagation and majority networks) and showed that this allowed for a clearer, deterministic model (Ishida, 1995).

Vertosick and Kelly noted that most antigenic elements present more than one epitope; since this may allow for better recognition properties, their model of the immune system attempts to capture this aspect of the natural immune system. Since they see the immune network as a form of neural network, they describe their immune network system as a multi-epitope recognizing neural network (Vertosick and Kelly, 1991).

The State-of-the-Art De Castro and von Zuben's CLONALG has been extended to become a network model by adding a network interaction phase after clonal selection has taken place. This network interaction removes B-cells that are close to other B-cells in the state space. Once the network has been allowed to develop, it is then converted to a minimum spanning tree, and the longest edges are removed. Finally, the resulting trees are analyzed to show the location of clusters.

De Castro has shown that AINET is able to efficiently compress cluster data; automatically find clusters, and can be applied in applications such as finding the center of radial basis functions (de Castro and Von Zuben, 2001; ?). He also compared AINET to a standard self-organizing map (SOMs) and showed that the results from AINET were far superior terms of quality and predictive value; however, it is not clear that it can be used in the same range of applications as SOMs.

The AINET method is interesting for its introduction of the network-to-tree conversion and tree analysis stages, and it suggests the question, would the method work without them? If not then is this still an immune algorithm? Would other means of producing a network be more or less effective?

Although Neal's work in providing a reliable means of stabilizing the Timmis-Neal network may stimulate work in the Timmis-Neal model (and indeed in AIRS) others take a different approach. Nasaroui, González and others have presented work in scalable, dynamic immune networks, which concentrates on their unsupervised learning abilities (Nasaroui et al., 2003), and other work explores the effects of fuzzifying immune networks (Nasaroui et al., 2002b).

Recent work combines the most recent elements of Timmis' and Neal's work to develop the Context-Aware Immune System (CAIS) (Mohr and Timmis, 2004). CAIS uses the data compression and generalization abilities of the network immune system, and has been applied to the problem of cleaning up global positioning satellite (GPS) data. Their results were very impressive and their method is currently the *only* way GPS data can be stripped of problem reflections, and outliers.

6.3 Typical Applications

Network models have been applied to:

- detecting gene promoter sequences (Cooke and Hunt, 1995);
- data mining (Hunt and Fellows, 1996);
- diagnosis (Ishida, 1993);
- cluster analysis (Timmis et al., 2000).

6.4 Applying the Usefulness Criteria

D.1-Symbol Distinctiveness: The use of vectors of binary or real-valued numbers is common in other methods. [D.1 = No]

D.2–Expression Distinctiveness, and D.3–Process Distinctiveness: Farmer has suggested that there is a general class of connectionist systems, with immune networks being just one example of such a system (Farmer, 1990). His analysis is almost precisely that which was advocated in Section 2—he sets out to “...*identify a common language across several fields in order to make their similarities and differences clearer.*”

Farmer’s analysis is mostly concerned with ANNs that exhibit Hebbian learning, but there also seems to be a distinct relationship between Timmis’ work on unsupervised clustering and self-organizing maps such as Kohonen’s LVQ (Kohonen, 1990). His remaining analysis will not be repeated here; however, he concludes that there is some distinctiveness in immune networks, and the intention of his analysis was to allow results to be transferred between the various connectionist approaches, particularly to help immune networks to grow as a field. The expressions and processes used in immune networks may be similar to other methods, but they are sufficiently different, Farmer says, that they should be pursued as a separate field.

Dasgupta has made a comparison between AIS and ANNs in which he highlights the similarities and differences in symbols, expressions and processes (Dasgupta, 1997).

Again, the ARB is quite an unusual data structure that has subtle effects on the dynamics of the network by removing the effects of multiple, identical instances of an antibody. [D.2. = No, except for the ARB; D.3 = Yes]

E.1–Does the Method Provide Unique Results?: No unique results appear to have been presented for immune networks. [E.1 = Yes (in one case, (Mohr and Timmis, 2004))]

E.2–Does the Method Produce Better Quality Results than Existing Methods?: (Cooke and Hunt, 1995) presented results for an early Timmis-Neal model, that showed it was as accurate as a backpropagation neural network when predicting gene promoter regions in DNA and, was only bettered by a method that was a hybrid of two other algorithms, although these results are now rather dated. The results of de Castro’s AINET are quite impressive, but more tests are required to demonstrate how widely it can be used. [E.2 = Yes, but newer, more extensive comparisons are required]

E.3–Is the Method Faster than Existing Methods?: Immune networks are usually slow. Most immune network methods have a time complexity of $O(N^2)$ because they make comparisons between each possible pair of antibodies in a set of antibodies. Some methods are emerging that attempt to reduce this complexity by allowing only a finite number of comparisons per ‘generation’, for example Garrett’s cellular automata styled ARPEN method (Garrett, 2003), but the tradeoff is that the results produced by the network are poorer due to the reduced number of network interactions. [E.3 = No]

7 Are AIS Useful?

Summary This section summarizes the results of applying the ‘usefulness criteria’ from Section 2 to each AIS method and then makes some final conclusions. DT-AIS will not be included in the assessment here, due to its underdeveloped status, although

it is hoped the discussion above was useful as an initial critique. The distinctiveness and effectiveness of the remaining AIS methods are summarized as in Table 1.

Table 1: The results of applying the ‘usefulness criteria’ to established types of AIS

Negative Detection Methods:			
D.1	No	E.1	Yes
D.2	No	E.2	In some cases perhaps
D.3	Yes	E.3	No
Overall : Yes		Overall : Yes	
Clonal Selection Methods:			
D.1	No	E.1	No
D.2	No, except for ARB	E.2	Sometimes, but superceded by AINET?
D.3	Yes	E.3	Not yet clear
Overall : Yes		Overall : Sometimes, but superceded by AINET	
Immune Network Methods:			
D.1	No	E.1	No
D.2	No, except for ARB	E.2	Yes, but newer, extensive comparisons required
D.3	Yes	E.3	Yes (at least one case)
Overall : Yes		Overall : Yes, but newer, extensive comparisons required	

Negative detection appears to be distinct, the closest analog probably being positive-only learning in ILP. Comparison with many more techniques is required, but this will come with time as the field of AIS matures. Although it has been shown that there are application areas in which negative detection is uniquely able to work, the evidence could be more convincing. The discovery of new application areas in which negative detection can triumph will not only strengthen its position, it will undoubtedly help its theoretical development—perhaps negative databases (Esponda and Forrest, 2004) will prove to be one example.

Clonal selection has also been presented as a distinct method, albeit one that is an immune-inspired form of GEC. The type of mutation, and method of reproduction, are both novel, as is the two-stage evaluation-selection operation per generation. In terms of effectiveness, CLONALG provides excellent results for optimization problems, but are they superior to other methods? This is still unclear. A comparison to GEC may help. GEC is widely studied despite there being deterministic methods that can optimize faster and better under some circumstances. GECs provide an excellent, general framework for optimization, and, furthermore, their study is worthwhile because investigating evolutionary processes is interesting, and can be simpler, than deterministic methods. These arguments also hold true for clonal selection, which is a form of GEC.

The distinctiveness of immune network methods is inherited from clonal selection, and has also been shown to be distinct from other, similar methods, such as classifier systems. Hunt and Cooke’s now dated work is still one of the most effective, although de Castro’s AINET may prove to be a better replacement for SOMs in a variety of applications, and Mohr and Timmis’ work on using network models for GPS data is notable.

Conclusion Perhaps the biggest difficulty faced by AIS is that it has few application types for which it is undisputedly the most effective method. Despite the many points in its favor, this single point is enough to allow it to be ignored by many. Although two important areas have been identified in which AIS is unique in its ability to provide solutions, further impressive demonstrations of effectiveness will be required if AIS is to be pushed to the fore.

In a related point, researchers in both negative (and positive) detection (Gonzalez and Dagupta, 2003; Gomez et al., 2003), and various types of immune network (de Castro and Von Zuben, 2001; Nasaroui et al., 2002b), are beginning to produce algorithms that are hybrids of AIS with other methods. This is perhaps an indication that generalist versions of current AIS methods alone are not powerful enough for some tasks, but it should not be taken to mean that tailored applications of AIS can not produce excellent results.

It is notable that there is a strong relationship between clonal selection and immune networks; in future, clonal selection may be classified as a form of network modeling with the degree of networking set to zero (Garrett, 2003). This would leave two main areas of AIS research, negative/danger detection and immune networks: one that focuses on detecting antigen or danger and the other that focuses on destroying it. The immune system itself has much more to offer than this: the innate immune system, for example, has been all but ignored, as has the immune system's use of parallel attack processes. As a whole, the immune system can be viewed as an adaptive, homeostatic control system (Somayaji, 2002; Bersini, 1991), and this suggests interesting possibilities, perhaps for a more unified AIS, and certainly for new applications in control.

In conclusion then, is AIS a useful paradigm? It has reliably provided three *distinct* types of method, and in most cases has produced highly *effective* results; however, these two aspects have not yet been *reliably* combined—therefore, in terms of the criteria in Section 2, the current answer appears to be, 'almost.' However, it is only now that AIS is beginning to mature: the annual ICARIS conferences, which began in 2002 to provide a forum exclusively for AIS research, have already provided a good deal of stimulation to the field; negative selection and danger theory appear to have a way forward from the 'detector impasse', and both clonal selection and immune networks are exploring new ground with encouraging results. It seems extremely likely that AIS will become increasingly useful over the next few years.

Acknowledgments

The author gratefully acknowledges informal contributions by Leandro de Castro, Jon Timmis and Mark Neal. Many thanks to Stephanie Forrest for her careful, generous comments on the original manuscript. Warm thanks also to Joanne Walker for her help throughout.

References

- Aickelin, U., Bentley, P., Cayzer, S., Kim, J., and McLeod, J. (2003). Danger theory: The link between AIS and IDS? In *2nd International Conference in Artificial Immune Systems (ICARIS-2003)*, pages 147–155, Edinburgh, UK.
- Aickelin, U. and Cayzer, S. (2002). The danger theory and its application to ais. In *Proceedings of the First International Conference on Artificial Immune Systems (ICARIS-2002)*, pages 141–148.
- Anchor, K. P., Zydallis, J. B., Gunsch, G. H., and Lamont, G. B. (2002). Extending the computer defense immune system: Network intrusion detection with a multiobjective evolutionary programming approach. In *Proceedings of the First International Conference on Artificial Immune Systems (ICARIS-2002)*.
- Ayara, M., Timmis, J., de Lemos, R., de Castro, L., and Duncan, R. (2002). Negative selection: How to generate detectors. In *Proceedings of the First International Conference on Artificial Immune Systems (ICARIS-2002)*, pages 89–98. University of Kent print unit.
- Bäck, T. and Schwefel, H.-P. (1993). An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1(1):1–23. This is an English version of several earlier German papers.

- Balthrop, J., Esponda, F., Forrest, S., and Glickman, M. (2002). Coverage and generalisation in an artificial immune system. In *Proceedings of GECCO-2002*, pages 3–10.
- Bersini, H. (1991). Immune network and adaptive control. Technical report, IRIDIA - Université Libre de Bruxelles, Bruxelles, Belgium.
- Bersini, H. (2002). Self-assertion versus self-recognition: A tribute to Francisco Varela. In *Proceedings of the First International Conference on Artificial Immune Systems (ICARIS-2002)*, pages 107–112. University of Kent print unit.
- Bersini, H. and Calenbuhr, V. (1997). Frustrated chaos in biological networks. *J. Theor. Biol.*, 188:187–200.
- Bradley, D. W. and Tyrell, A. M. (2002). A hardware immune system for benchmark state machine error detection. In *Congress on Evolutionary Computation. Part of the World Congress on Computational Intelligence*, pages 813–818.
- Burgess, M. (1998). Computer immunology. In *Proceedings of the 12th Systems Administration Conference (LISA-1998)*, Boston, USA.
- Burnet, F. M. (1959). *The Clonal Selection Theory of Acquired Immunity*. Cambridge University Press.
- Castiglione, F., Motta, S., and Nicosia, G. (2001). Pattern recognition with a multi-agent model of the immune system. In *International NAISO Symposium on Information Science Innovations in Engineering of Natural and Artificial Intelligent Systems (ENAIIS'2001)*, The American University in Dubai, U.A.E. ICSC Academic Press.
- Ceong, H.-T., Kim, Y., Lee, D., and Lee, K.-H. (2003). Complementary dual detectors for effective classification. In *2nd International Conference in Artificial Immune Systems (ICARIS-2003)*, pages 242–248, Edinburgh, UK.
- Chowdhury, D. and Stauffer, D. (1992). Statistical physics of immune networks. *Physica A*, 186:61–81.
- Chowdhury, D., Stauffer, D., and Choudary, P. (1990). A unified discrete model of immune response. *Journal of theoretical Biology*, 145:207–215.
- Cleland, W. W. (1963). The kinetics of enzyme-catalysed reactions with two or more substrates and products: 1. Nomenclature and rate equations. *Biochem. Biophys. Acta*, 67:104–137.
- Cobb, H. (1990). An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time dependent nonstationary environments. NRL Memorandum Report 6760 (NCARAI report AIC-90-001).
- Cooke, D. and Hunt, J. (1995). Recognising promoter sequences using an artificial immune system. In *Proceedings of the 3rd International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 89–97.
- Cutello, V. and Nicosia, G. (2002). Multiple learning using immune algorithms. In *4th International Conference on Recent Advances in Soft Computing (RASC-2002)*, pages 102–107, Nottingham, UK.
- Dasgupta, D. (1997). Artificial neural networks and artificial immune systems: Similarities and differences. In *The proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Orlando.
- Dasgupta, D. (1999). An overview of artificial immune systems and their applications. In *Artificial Immune Systems and Their Applications*, chapter 1, pages 1–21. Springer-Verlag, Inc.
- Dasgupta, D. and Forrest, S. (1995). Tool breakage detection in milling operations using a negative selection algorithm. Technical Report Report no. CS95-5, Department of Computer Science, University of New Mexico.

- Dasgupta, D. and Forrest, S. (1999). An anomaly detection algorithm inspired by the immune system. In *Artificial Immune Systems and Their Applications*, chapter 14, pages 262–277. Springer-Verlag, Inc.
- Dasgupta, D. and Nino, F. (2000). A comparison of negative and positive selection algorithms in novel pattern detection. In *The Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC)*, Nashville.
- Dasgupta, D. and Yu, S. (2003). Artificial immune systems a bibliography. Technical Report CS-03-002, University of Memphis, Department of Computer Science.
- Dawkins, R. (1976). *The Selfish Gene*. Oxford University Press, Oxford, UK.
- De Boer, R. J., Kevrekidis, I. G., and Perelson, A. S. (1992). Immune network behavior II: From oscillations to chaos and stationary states. Technical Report 95-05-024.
- de Castro, L. N. and Von Zuben, F. J. (2001). An immunological approach to initialize centers of radial basis function neural networks. In *Proc. of 5th Brazilian Conference on Neural Networks*, pages 79–84.
- de Castro, L. N. and Von Zuben, F. J. (1999). Artificial immune systems: Part I—basic theory and applications. Technical Report DCA-RT 01/99, School of Computing and Electrical Engineering, State University of Campinas, Brazil.
- de Castro, L. N. and Von Zuben, F. J. (2000). The clonal selection algorithm with engineering applications. In *Proceedings of GECCO'00, Workshop on Artificial Immune Systems and Their Applications*, pages 36–37.
- de Castro, L. N. and Von Zuben, F. J. (2001). AINET: An artificial immune network for data analysis. In Abbass, H. A., Sarker, R. A., and Newton, C. S., editors, *Data Mining: A Heuristic Approach*, chapter XII, pages 231–259. Idea Group Publishing, USA.
- de Castro, L. N. and Von Zuben, F. J. (2002). Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6(3):239–251.
- Deaton, R., Garzon, M., Rose, J. A., Franceschetti, D. R., Murphy, R. C., and Jr., S. E. S. (1997). DNA based artificial immune system for self-nonsel self discrimination. In *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*. IEEE.
- D'haeseleer, P. (1996). An immunological approach to change detection: Theoretical results. In *Proceedings of the 9th IEEE Computer Security Foundations Workshop*, pages 132–143. IEEE Computer Society Press.
- D'haeseleer, P., Forrest, S., and Helman, P. (1996). An immunological approach to change detection: Algorithms, analysis and implications. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 132–143. IEEE Computer Society Press.
- Dorigo, M. (1992). *Optimization, learning and natural algorithms*. PhD thesis, DEI, Politecnico di Milano, Italia. In Italian.
- Dorigo, M. (1999). Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137–172.
- Ebner, M., Breunig, H.-G., and Albert, J. (2002). On the use of negative selection in an artificial immune system. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002)*, pages 957–964, New York, NY. Morgan Kaufman Publishers, San Francisco, CA.
- Esponda, F. and Forrest, S. (2003). The crossover closure and partial match detector. In *2nd International Conference in Artificial Immune Systems (ICARIS-2003)*, pages 249–260, Edinburgh, UK.
- Esponda, F. and Forrest, S. (2004). Online negative databases. In *3rd International Conference in Artificial Immune Systems (ICARIS-2004)*, pages 175–188, Catania, Italy.

- Esponda, F., Forrest, S., and Helman, P. (2003). A formal framework for positive and negative detection schemes. *IEEE Trans. on Systems, Man and Cybernetics Part B: Cybernetics*, page In Press.
- Esponda, F., Forrest, S., and Helman, P. (2004). A formal framework for positive and negative detection. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(1):357–373.
- Farmer, J. (1990). A rosetta stone for connectionism. *Physica D*, 42:153–187.
- Farmer, J., Packard, N., and Perelson, A. (1986). The immune system, adaptation and machine learning. *Physica D*, 22:187–204.
- Forrest, S. and Hofmeyr, S. (2000). Immunology as information processing. In Segel, L. and Cohen, I., editors, *Design Principles for Immune System and Other Distributed Autonomous Systems*. Oxford University Press.
- Forrest, S., Javornik, B., Smith, R. E., and Perelson, A. S. (1993). Using genetic algorithms to explore pattern recognition in the immune system. *Evolutionary Computation*, 1(3):191–211.
- Forrest, S., Perelson, A. S., Allen, L., and Cherukuri, R. (1994). Self-nonsel self discrimination in a computer. In *Proceedings of 1994 IEEE Symposium on Research in Security and Privacy*, pages 132–143.
- Freitas, A. A. and Timmis, J. (2003). Revisiting the foundations of artificial immune systems: A problem-oriented perspective. In *2nd International Conference in Artificial Immune Systems (ICARIS-2003)*, pages 229–241, Edinburgh, UK.
- French, D. L., Reuven, L., and Scharff, M. D. (1989). The role of somatic hypermutation in the generation of antibody diversity. *Science*, 244(4909):1152–1157.
- Fukuda, T., Mori, K., and Tsukiyama, M. (1993). Immune networks using genetic algorithm for adaptive production scheduling. In *15th IFAC World Congress*, volume 3, pages 57–60.
- Fukuda, T., Mori, K., and Tsukiyama, M. (1998). Parallel search for multi-modal function optimization with diversity and learning of immune algorithm. In Dasgupta, D., editor, *Artificial Immune Systems and their Applications*, pages 210–220. Springer-Verlag, Berlin.
- Garrett, S. (2004). Parameter-free clonal selection. In *Proceedings of the Congress of Evolutionary Computation*.
- Garrett, S. M. (2003). A paratope is not an epitope: Implications for immune networks and clonal selection. In *2nd International Conference in Artificial Immune Systems*, pages 217–228, Edinburgh. Springer-Verlag.
- Gaspar, A. and Collard, P. (1999). From GAs to artificial immune systems: Improving adaptation in time dependent optimization. In *Proceedings of the Congress on Evolutionary Computation*, volume 3, pages 1859–1866, Mayflower Hotel, Washington D.C., USA. IEEE Press.
- Gomez, J., Gonzalez, F., and Dasgupta, D. (2003). An immuno-fuzzy approach to anomaly detection. In *Proceedings of the 12th IEEE International Conference on Fuzzy Systems (FUZZIEEE)*, volume 2, pages 1219–1224.
- González, F. and Dasgupta, D. (2002). An immunogenetic technique to detect anomalies in network traffic. In *Proceedings of GECCO-2002*, pages 1081–1088.
- Gonzalez, F. and Dagupta, D. (2003). Anomaly detection using real-valued negative selection. *Genetic Programming and Evolvable Machines*, 4(4):383–403.
- Goodman, D., Boggess, L., and Watkins, A. (2002). Artificial immune system classification of multiple-class problems. In Dagli, C., Buczak, A., Ghosh, J., Embrechts, M., Ersoy, O., and Kercel, S., editors, *Intelligent Engineering Systems Through Artificial Neural Networks, Fuzzy Logic, Evolutionary Programming Complex Systems and Artificial Life*, volume 12, pages 179–184. ASME Press, New York.

- Greensmith, J. and Cayzer, S. (2003). An artificial immune system approach to semantic document classification. In *2nd International Conference in Artificial Immune Systems (ICARIS-2003)*, pages 136–146, Edinburgh, UK.
- Hart, E. and Ross, P. (2002). Exploiting the analogy between immunology and sparse distributed memories: A system for clustering non-stationary data. In *Proceedings of the First International Conference on Artificial Immune Systems (ICARIS-2002)*, pages 49–58, Canterbury, UK.
- Hart, E., Ross, P., and Nelson, J. (1998). Producing robust schedules via an artificial immune system. In *IEEE World Congress on Computational Intelligence, International Conference on Evolutionary Computing*.
- Helman, P. and Forrest, S. (1994). An efficient algorithms for generating random antibody strings. Technical Report CS-94-07, Department of Computer Science, University of New Mexico.
- Hightower, R., Forrest, S., and Perelson, A. S. (1996). The Baldwin effect in the immune system: Learning by somatic hypermutation. In Belew, R. K. and Mitchell, M., editors, *Adaptive Individuals in Evolving Populations: Models and Algorithms*, pages 159–167. Addison Wesley, Reading, MA.
- Hofmeyr, S. and Forrest, S. (2000). Architecture for the artificial immune system. *Evolutionary Computation*, 8(4):443–473.
- Hofmeyr, S. and Forrst, S. (1999). Immunity by design: An artificial immune system. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, pages 1289–1296.
- Hunt, J. and Fellows, A. (1996). Introducing an immune response into a CBR system for data mining.
- Ishida, Y. (1993). An immune network model and its applications to process diagnosis. *Systems and Computers in Japan*, 24(6):646–651.
- Ishida, Y. (1995). Note on immune network and majority network. Technical Report NAIST-ISTR95031, Graduate School of Information Science, Nara Institute of Science and Technology.
- Ishiguru, A., Watanabe, Y., and Uchikawa, Y. (1994). Fault diagnosis of plant systems using immune networks. In *Proceedings of the 1994 IEEE International Conference on Mutisensor Fusion and Integration for Intelligent Systems (MFI'94)*, pages 34–42.
- Jerne, N. (1974). Towards a network theory of the immune system. *Annals of Immunology*, 125:373–389.
- Kim, J. and Bentley, P. J. (2001). An evaluation of negative selection in an artificial immune system for network intrusion detection. In *Proceedings of GECCO-2001*, pages 1330–1337, San Francisco, USA.
- Knight, T. and Timmis, J. (2001). AINE: An immunological approach to data mining. In Cercone, N., Lin, T., and Wu, X., editors, *IEEE International Conference on Data Mining*, pages 297–304, San Jose, CA. USA.
- Kohonen, T. (1990). The self-organizing map. *Procs. IEEE*, 78:1467 ff.
- Lane, T. and Brodley, C. E. (1997). Sequence matching and learning in anomaly detection for computer security. In Fawcett, Haimowitz, Provost, and Stolfo, editors, *AI Approaches to Fraud Detection and Risk Management*, pages 43–49. AAAI Press.
- Lau, T. L. and Tsang, E. P. K. (1996). Applying a mutation-based genetic algorithm to processor configuration problems. In *Proceedings of the 8th IEEE Conference on Tools with Artificial Intelligence (ICTAI'96)*, Toulouse, France.
- Matzinger, P. (2002). The Danger Model: A renewed sense of self. *Science*, 296(5566):301–305.

- McDermott, D. (1981). Artificial intelligence meets natural stupidity. In Haugeland, J., editor, *Mind Design: Philosophy, Psychology, Artificial Intelligence*, pages 143–160. MIT Press, Cambridge, MA.
- Medzhitov, R. and Janeway, C. A. (2002). Decoding the patterns of self and nonself by the innate immune system. *Science*, 296(5566):298–300.
- Minsky, M. L. and Papert, S. A. (1969). *Perceptrons*. MIT Press, Cambridge, MA.
- Mohr, P. and Timmis, J. (2004). Exploiting immunological properties for ubiquitous computing systems. In *3rd International Conference in Artificial Immune Systems (ICARIS-2004)*, pages 277–289, Catania, Italy.
- Mori, K., Tsukiyama, M., and Fukada, T. (1993). Immune algorithm with searching diversity and its application to resource allocation problem. *Trans. of the Institute of Electrical Engineers of Japan*, 113-C(10):872–878. In Japanese.
- Muggleton, S. (1996). Learning from positive data. In Muggleton, S., editor, *Proceedings of the 6th International Workshop on Inductive Logic Programming*, pages 358–376. Springer-Verlag.
- Nasaroui, O., Gonzalez, F., Cardona, C., Rojas, C., and Dasgupta, D. (2003). A scalable artificial immune system model for dynamic unsupervised learning. In *The proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, Chicago.
- Nasaroui, O., González, F., and Dasgupta, D. (2002a). The fuzzy artificial immune system: Motivations, basic concepts, and application to clustering and web profiling. In *IEEE International Conference on Fuzzy Systems*, pages 711–716, Hawaii, HI.
- Nasaroui, O., Gonzalez, F., and Dasgupta, D. (2002b). The fuzzy artificial immune system: Motivations, basic concepts, and application to clustering and web profiling. In *IEEE International Conference on Fuzzy Systems, Congress on Computational Intelligence*, volume 1, pages 711–716, Hawaii.
- Neal, M. (2002). An artificial immune system for continuous analysis of time-varying data. In *Proceedings of the First International Conference on Artificial Immune Systems (ICARIS-2002)*, pages 76–85. University of Kent print unit.
- Neal, M. (2003). Meta-stable memory in an artificial immune network. In *2nd International Conference in Artificial Immune Systems (ICARIS-2003)*, pages 168–180, Edinburgh, UK.
- Newell, A. and Simon, H. (1976). Computer science as empirical enquiry: Symbols and search. *Communications of the ACM*, 19(3).
- Nicosia, G., Cutello, V., and Pavone, M. (2003). A hybrid immune algorithm with information gain for the graph coloring problem. In *Genetic and Evolutionary Computation Conference (GECCO-2003) LNCS 2723*, pages 171–182, Chicago, Illinois, USA.
- Nikolaev, N. I., Iba, H., and Slavov, V. (1999). Inductive genetic programming with immune network dynamics. In *Advances in Genetic Programming 3*, chapter 15. MIT Press.
- Parthasarathy, K. (2003). Clonal selection method for immunity-based intrusion detection system. Location: <http://web.umr.deu/tauritzd/courses/cs447/project/>, also available from SMG.
- Paun, G. (2002). *Membrane Computing: An Introduction*. Springer Verlag.
- Percus, J. K., Percus, O. E., and Perelson, A. S. (1993). Predicting the size of the antibody combining region from consideration of efficient self-nonsel self discrimination. *Proceedings of the National Academy of Science*, 90:1691–1695.
- Perelson, A. S. and Weisbuch, G. (1997). Immunology for physicists. *Rev. Modern Phys.*, 69:1219–1267.

- Rechenberg, I. (1994). Evolution strategy. In et.al., Z., editor, *Computational Intelligence: Imitating Life*, chapter 14, pages 147–159. IEEE Press.
- Secker, A., Freitas, A. A., and Timmis, J. (2003). A danger theory inspired approach to web mining. In *2nd International Conference in Artificial Immune Systems (ICARIS-2003)*, pages 156–167, Edinburgh, UK.
- Smith, D. J., Forrest, S., and Perelson, A. S. (1996). Immunological memory is associative. In *Int. Conference of Multiagent Systems*, pages 62–70, Kyoto, Japan. Workshop Notes, Workshop 4: Immunity Based Systems.
- Somayaji, A. (2002). *Operating system stability and security through process homeostasis*. PhD thesis, University of New Mexico.
- Stadler, P. F., Schuster, P., and Perelson, A. S. (1994). Immune networks modelled by replicator equations. *J. Math. Biol.*, 33:111–137.
- Timmis, J., Neal, M., and Hunt, J. (2000). An artificial immune system for data analysis. *Biosystems*, 55(1/3):143–150.
- Timmis, J. and Neal, M. J. (2000). A resource limited artificial immune system for data analysis. In *Proceedings of ES2000*, pages 19–32, Cambridge, UK.
- Vertosick, F. and Kelly, R. (1991). The immune system as a neural network: A multi-epitope approach. *Journal of Theoretical Biology*, 150:225–237.
- Walker, J. H. and Garrett, S. M. (2003). Dynamic function optimisation: Comparing the performance of clonal selection and evolution strategies. In *2nd International Conference in Artificial Immune Systems (ICARIS-2003)*, pages 273–284, Edinburgh, UK.
- Watkins, A. and Timmis, J. (2002). Artificial immune recognition system (AIRS): Revisions and refinements. In Timmis, J. and Bentley, P., editors, *1st International Conference on Artificial Immune Systems*, pages 173–181. University of Kent Print Unit.
- Weinand, R. G. (1990). Somatic mutation, affinity maturation and antibody repertoire: A computer model. *J. of Theor. Biol.*, 143:343–382.
- White, J. A. and Garrett, S. M. (2003). Improved pattern recognition with artificial clonal selection. In *Proceedings of the Second International Conference on Artificial Immune Systems (ICARIS-03)*, pages 181–193.
- Wierchoń, S. T. (1999). Generating antibody strings in an artificial immune system. Technical Report ICS PAS Report No. 892, Institute of Computer Science, Polish Academy of Sciences.
- Wierchoń, S. T. (2000). Discriminative power of the receptors activated by k -contiguous bits rule. *Journal of Computer Science and Technology*, 1(3):1–13.
- Wierchoń, S. T. (2002). Function optimization by the immune metaphor. *Task Quarterly*, 6(3):1–16.
- Williamson, M. M. (2002). Biologically-inspired approaches to computer security. Technical Report HPL-2002-131, HP Labs, Bristol, UK. Available from <http://www.hpl.hp.com/techreports/2002/HPL-2002-131.html>.
- Wolpert, D. H. and Macready, W. G. (1995). No free lunch theorems for search. Technical Report SFI-95-02-010, Santa Fe Institute.