Computer Science at Kent

Assessing the performance of the resource limited artificial immune system AINE

Thomas Knight and Jon Timmis

Technical Report No. 3-01
March 2001

Published by the Computing Laboratory,
University of Kent, Canterbury, Kent CT2 7NF, UK

Abstract

This report presents an assessment of the resource limited artificial immune system known as AINE. This work is a continuation of previous work to develop an artificial immune system for data analysis. A brief introduction to the fundamentals of immunology is given followed by a review of the previous work on AINE. Discrepancies in the original work are identified and revisions are made. Results are then presented for a simulated data set and for the Fisher Iris data set. Comparisons are then drawn between the revised algorithm and the original version of AINE. Results of parameter adjustment on the revised version confirm those results in the previous work. Additionally, trends in the evolution of the networks are identified and analysed. These tests show new behaviour in the revised AINE and from which it can be concluded that AINE no longer shows the possibility of continual learning, but exhibits characteristics of optimisation, which are identified for future study.

1 Introduction

The human immune system can be seen as a complex network structure that is capable of adapting to and learning about foreign invaders such as bacteria or viruses. It is these properties that have interested computer scientists and driven them to produce various tools that use metaphors extracted from the immune system. In this situation a metaphor can be seen as a generalisation of behaviour or a complex reaction. An example of one such tool is AINE (Timmis, 2000a); metaphors dealing with the structure and learning behaviour of the immune system are taken and applied in an algorithm that is capable of learning and representation key features of data sets, it is also suggested that there is the possibility that once data is learnt from one data set, the learning can continue to incorporate new information from the same domain.

This report presents a re-implementation of the Artificial Immune Network originally proposed in (Timmis and Neal, 2000). An unsupervised learning algorithm inspired by the human immune system, called AINE. It was proposed that AINE had the ability to produce networks that can identify patterns or clusters in multi-dimensional data. The original scope of this report was to investigate and test AINE and to conduct further testing of on a wider range of data sets than in previous work (Timmis, 2000b). After re-implementation discrepancies were identified in AINE and revisions were made to correct these problems. These revisions and their impacts are discussed in this report.

The revised version of AINE is tested using the same suite of test criteria as used in (Timmis, 2000b) and tested on two data sets; a simulated data set and the Iris data set (Fisher, 1936). Additionally the evolution of the networks are considered and trends in the evolution identified and analysed. The results are compared to the previous work and conclusions are drawn.

Results of AINE show that it can be used for one-shot learning, but over time weaker patterns are lost from the networks and only the stronger remain. Therefore continual learning with this algorithm is not possible.

2 Previous Work

Over the last two decades understanding of the human immune system has significantly advanced. Scientist now believe that they have a good understanding of how the immune system work and how it defends our bodies from attack by foreign invaders (such as viruses and bacteria). The immune system has shown that it is capable of remembering previous encounters with these invaders and this knowledge has been harnessed to provide vaccines for a whole plethora of viruses and diseases. It also shows highly distributed detection and memory systems, diversity of detection ability across individuals, inexact matching strategies, and sensitivity to most new foreign patterns (Forrest *et al*., 1996). To computer scientists this ability to recognise and remember encounters with invaders over long periods of time is of great interest. These mechanisms have been extracted by computer scientist over the last decade for work on data mining (Hunt and Fellows, 1996), intrusion detection (Dasgupta, 1999) and data analysis (Timmis, 2000a).

The resource limited artificial immune system proposed in (Timmis, 2000a), called AINE, uses high-level metaphors extracted from the human immune system to analyse multi-dimensional data sets. The aims being to produce networks that describe the key features of data items within the set. AINE is modelled on the interactions of B cells and antigens and

uses the immune network theory (Jerne, 1974 and Perelson, 1986) as a source of inspiration. A full description of AINE and the metaphors used can be found in (Timmis and Neal, 2000).

An introduction to the human immune system will now be presented and a brief summary of the immune network theory will be given and why metaphors taken from the immune system can be used in data mining and machine learning. The main features of AINE will be summarised and how these relate to the immune system.

2.1 Immunology

The following is a summary of the immune system as described in (Nossal, 1994). The immune system is a very complex "hunt and destroy" mechanism that works at the cellular level in our bodies. The task of the immune system is to identify and destroy foreign invaders or *antigens*. Antigens can be though of as bacteria, viruses and fungi that may cause disease. The immune system is composed of *lymphocytes,* which are white blood cells whose task it is to identify and destroy invading antigens. There are two types of lymphocyte, *B lymphocytes* (B-cells) and *T lymphocytes* (T-cells). B-cells are lymphocytes that have matured in the bone marrow and these produce the *Antibodies* that bind to the invading antigens and help destroy them. T-cells mature in the thymus and there are two types, *Helper T lymphocytes* regulate and control the strength of the immune response. *Killer T lymphocytes* directly destroy cells that have specific antigens on their surface that are recognised by these T-cells. Antibodies are Y-shaped proteins that are only made by B-cells (Figure 2.1). The antibody binds to the antigenic receptors at the ends of the arms of the Y. The base of the Y determines how the antigen will be destroyed. Lymphocytes can make billions of different kinds of antigenic receptors, each individual lymphocyte makes only one kind. When an antigen enters the body, it activates only the lymphocytes whose receptors can bind to it. When an antigen enters a cell, transport molecules (made from a group of genes called *major histocompatibility complex (MHC)*) attach themselves to the antigen and move them to the surface of the cell where they present the antigen to T-cells. *Class I MHC molecules* present antigens to Killer T-cells and *class II MHC molecules* present antigens to Helper T cells.

2.1.1 Immunity: Innate and Adaptive

There are two types of immunity: ***innate (or non-specific) immunity*** is the body's first line of defence against infection. Physical barriers such as *tears, skin, mucus and saliva* provide this. These provide a barrier mechanism that hinders the entrance of diseases, but does not destroy them. ***Adaptive (or specific) immunity*** is the main line of defence and has four key properties.

- Responds only if an invader is present
- It is specifically tailored to that invader
- It remembers previous contact with an invader, therefore responding faster after the first exposure
- It distinguishes between the self (body cells and tissue) and the non-self (antigens) (Taranakov & Dasgupta, 2000).

2.1.2 B Cells and their Antibodies

The B-cell is an integral part of the immune system. Through a process of recognition and stimulation, B-cells will clone and mutate to produce a diverse set of antibodies in an attempt to remove the infection from the body (Timmis, 2000). The antibodies are specific proteins that recognize and bind to another protein. The production and binding of antibodies is usually a way of signalling other cells to kill, ingest or remove the bound substance (de Castro, 1999). Each antibody has two paratopes and two epitopes that are the specialised part of the antibody that identify other molecules (Hunt & Cooke, 1996). Binding between antigens and antibodies is governed by how well the paratopes on the antibody matches the

epitope of the antigen, the closer this match, the stronger the bind. Although it is the antibody proteins that surround the B cell (Figure 2.1) that are responsible for recognising and attaching to antigen invaders, it is the B cell itself that has one of the most important roles in the immune system. When an antibody binds to an antigen the B cell becomes stimulated. The level of stimulation is a product of how well the antibody matches the antigen. If the stimulation level rises above a given threshold, the B cell undergoes rapid expansion and starts rapid cellular division resulting in a number of clones (Biologists are still unsure exactly how many clones this process produces) as shown in figure 2.1. To allow the immune system to be adaptive, the clones turn on a mutation mechanism known as *somatic hypermutation* (Kepler & Perelson, 1993). However, if the B cell fails to match the antigen closely enough, or any other antigenic molecules within the system, the cell will not remain stimulated and will eventually die off.

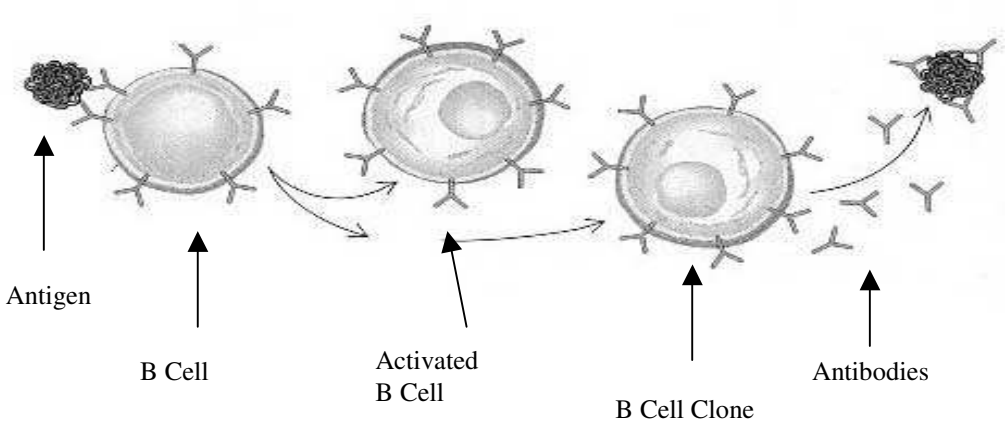


Figure 2.1: The B-cells interacts with an antigenic substance, becomes stimulated (activated) and clones producing thousands of antibodies. Adapted from (Nossal, 1994).

2.1.3 Primary and Secondary Response

The primary immune response occurs when the immune system encounters the antigenic agent for the first time and reacts against it. On encountering the antigenic agent the immune system will produce antibodies that can bind to the agent and assist in notifying either B cells or T cells to destroy it. The most popular theory about what happens next is that the immune system "learns" about the antigen by replicating matching antibodies and therefore preparing for the eventuality that they will meet again.

The secondary immune response occurs when the same antigen is encountered again. It is characterised by more rapid and more abundant production of antibody's resulting from the primary response (Hunt & Cooke, 1997).

2.1.4 Immune Network Theory

The immune network theory was first proposed by (Jerne, 1974) and reviewed by (Perelson, 1989). The theory states that the immune system can be though of as a network of B-cells. This network dynamically maintains the immune memory using feedback mechanisms. These feedback mechanisms are produced by the stimulation and suppression of neighbouring cells in the network. Without frequent stimulation B-cells can be lost or "forgotten" from the network, which highlights the importance of frequent immunisation. The idea of clonal selection is also proposed where B-cells can undergo a process of cloning and mutation to

allow the immune system to not only recognise the antigens currently causing a response, but also similar antigens. This provides the immune system with the capability to defend itself against invaders that are capable of mutation themselves.

2.1.5 Shape Space

The idea of antibody repertoire completeness has been postulated by (Coutinho, 1980) and summarised by (Perelson, 1989). Repertoire completeness is the immune systems ability to recognise all antigens and can be represented by the idea of shape space. The immune system of a given person can be represented by a two dimensional circle of volume *V* (Figure 2.2). This circle represents the finite number of gene combinations possible on an antibodies paratopes. Each antibody (*A*) can recognise a given number of genetic combinations and therefore can recognise a volume (V_e) of antigenic epitopes (x) in shape space. Therefore it is conceivable that the repertoire of antibodies can be deemed complete if they cover the entire volume of the shape space.

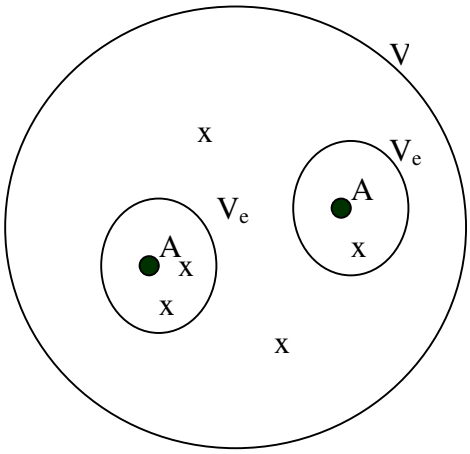


Figure 2.2: Adapted from (Perelson, 1989). A diagrammatic representation of shape space.

2.2 AINE

Data mining (part of Knowledge Discovery in Databases) is a rapidly emerging field of research and applications (Fayyad *et al*, 1996). It relates to the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data (Hunt and Fellows, 1996). Data mining is the application of the chosen model on the data set in order to extract useful and interesting patterns in the data. These include, but are not limited to numerical and conceptual clustering and classification (Timmis *et al*, 1999). A new technique in the field of data mining is Artificial Immune Systems (AIS). AIS's have been applied to a number of data mining problems, with a great deal of success. One such AIS, was proposed by (Timmis, 2000a) and a summarised description of that system, known as a resource limited artificial immune system or AINE is presented here and a review of which can be seen in (Timmis and Knight, 2001).

AINE employs a number of high-level metaphors drawn from the immune system. These are:
- A B-cell is capable of recognising pathogens.
- There are links between similar B-cells, and these links form a network of B-cells.
- Cloning and Mutation operations are performed on B-cells.
- A number of B-cells can be represented by an ABR given the theory of shape space (see section 2.1.5).

In the natural immune system, pathogens produce antigens when invading a host. It is these antigens that are matched with the antibodies of the immune system. For the sake of simplicity in AINE, separate antigens are not created; rather the complete data items are considered to be representative of antigens rather than entire pathogens.

AINE consists of a set of Artificial Recognition Balls (ARBs, Figure 2.4) and links between those ARBs, indicating similarities between them and is referred to as a network. ARBs compete for the ability to represent a number of B cells within AINE, based on stimulation of the ARB. The higher the ARB stimulation, the more B cells it can represent. Once an ARB no longer represents any B cells, it is removed from the network. Cloning and mutation mechanisms are present to introduce diversity in the network. A termination condition exists where AINE can be terminated if a period of stability is reached (which usually indicates when only the core network is present) or it can learn for a fixed number of iterations. A flow diagram showing the AINE algorithm can be seen in Figure 2.3. A full explanation of AINE is undertaken in the following subsections.

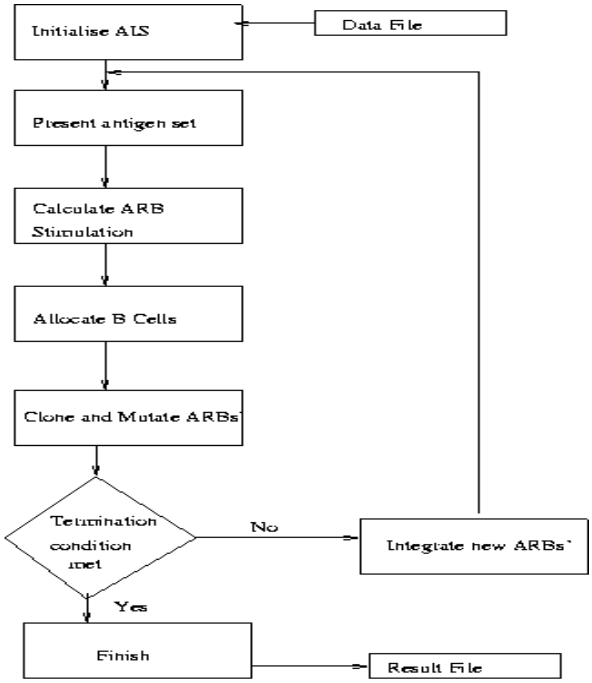


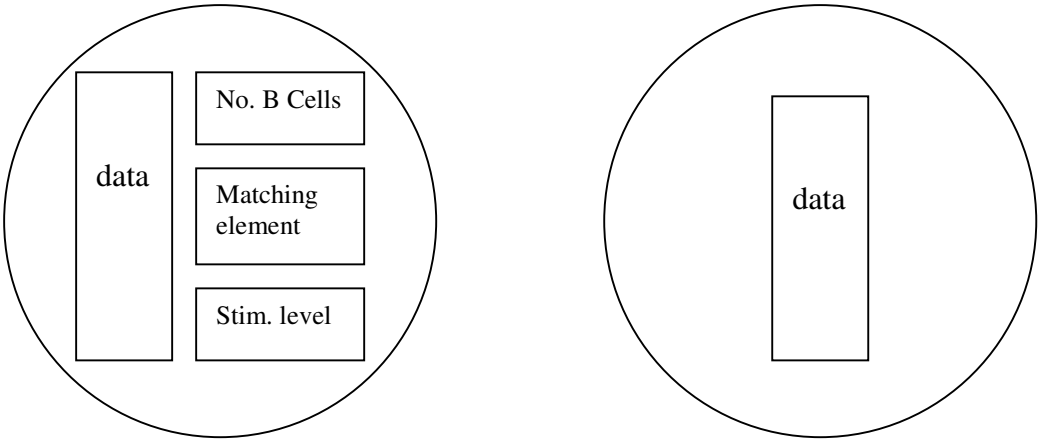
Figure 2.3: The AINE algorithm

2.2.1 Algorithm

AINE contains a population of ARBs that are used to store individual data items being analysed.

Raw data is used to create an initial population of ARBs objects and a set of training data objects (Figure 2.4). The initial population is a sample of the raw data set and the training data is the remainder of the set. AINE is then initialised by binding all ARB objects to each other in the initial population. This forms a network of ARBs that are linked where a link between

two ARBs is only formed if the affinity between them is below the network affinity threshold (NAT) and the two ARBs are not the same object.



(a) Internal view of the ARB object (b) Internal view of an antigen

Figure 2.4: How ARBs and Antigens are represented in AINE adapted from (Timmis, 2000a)

The network is evolved by repeatedly presenting the training data items to the ARBs. When a training data item is presented to an ARB the two data components are matched and the affinity (or Euclidean Distance) is calculated. This affinity is then considered to be the stimulation contributed by that data item. Each data item in the training set is presented to every ARB in the network and the stimulation of an ARB is the sum of all the stimulations contributed by the training set.

2.2.1.1 Parameters

There are four key parameters used in AINE, these are the *network affinity threshold* (NAT), the *mutation rate*, the *number of resources* and the *number of clones*.

- ***Network Affinity Threshold:*** The NAT is applied as a scalar value in the range of 0.0 to 1.0 and calculated using the following equation:

$$NAT = aff_{i,t}.ns \tag{1}$$

  where $aff_{i,t}$ is the average affinity between the initial population and the training population. *ns* is the NAT scalar. Previous work has show the NAT affects the connectivity of the networks produced. A high NAT produces very highly connected networks, and a low NAT will produce sparse networks.
- ***Mutation Rate:*** The mutation rate is also in the range 0.0 and 1.0 and defines the probability that a clone will mutate. This has been shown to control the diversity of the ARBs in the resultant networks and has no affect on the network connectivity and size.
- ***Number of Resources:*** The number of resources defines the maximum number of resources that can be claimed by all of the ARBs in the network. This value limits the population size and prevents the population explosion seen in previous work (Timmis and Neal, 2000). The number of resources is directly proportional to the network size and that has little or no effect on the connectivity of the network.
- ***Number of clones:*** The number of clones defines the maximum number of clones an ARB can produce if cloned and therefore contributes to the diversity of the networks produced.

2.2.1.2 ARB Objects

Each ARB represents a single piece of n-dimensional data and represents a number of identical B cells. This can be though of as the center of a ball in shape space (Figure 2.2) from (Perelson, 1989). An ARB my be stimulated by one or more antigens, where a stimulating antigen is that which matches the data item above the NAT. Stimulation and suppression may also come from similar ARB in the network. The total stimulation for an ARB is calculated using equation 2:

$$sl = \sum_{x=0}^{a}(1 - pd) + \sum_{x=0}^{n}(1 - dist_x) - \sum_{x=0}^{n}(dist_x) \tag{2}$$

where *pd* is defined as the distance between the ARB and the antigen in the normalised data space, such that $0 \leq pd \geq 1$, and dis_x the distance of the *x*th neighbour from the ARB.

2.2.1.3 Resource allocation

Each ARB is allocated resources based on its stimulation level. Allocation is according to the following equation:

$$R_i = k.(sl_i^2) \tag{3}$$

where R_i is the number of resources to allocate and *k* is a constant. This allocation mechanism allows a higher number of resources to be allocated to the most stimulated ARBs and a limited number to be allocated to the least stimulated.

Following the presentation of the antigen set and the calculation of each ARBs stimulation levels, the resources are allocated to all of the ARBs. If the total number of resources allocated is greater than the maximum number allowed, the difference is purged from the network using the algorithm in figure 2.5.

```
Allocate B cells to ARB's based on stimulation level
Calculate number of B cells over allocated
if too many B cells have been allocated
      repeat
            get the weakest ARB
            if number of resources held by ARB is zero
                  Remove ARB from the network
            if the number of B cells to remove > number B cells
held by ARB
                  Remove ARB
                  Decrement number of B cells to remove by
number held by ARB
            else
                  Decrement ARB by number of B cells
            endif
      until number over allocated equals zero
else
      Go through network and remove all ARB with zero B cells
endif
```

Figure 2.5: The resource allocation mechanism

This mechanism removes the weakest ARBs from the network until the total number of resources allocated is the same as the maximum number available.

2.2.1.4 Cloning and Mutation

Like the human immune system, only the most stimulated ARBs in the network clone. An ARB will clone if its stimulation level is greater that a randomly generated number and the number of clones it will produce is based on the following equation:

$$n = sl.mr \tag{4}$$

where n is the number of clones produced, sl is the stimulation level of the ARB, and mr is the number of clones. Each clone then has a chance to be mutated and this is represented by the mutation rate parameter. If a clone mutates, then the actually mutation is of a stochastic nature and one or more of the data items dimensions may be mutated. Only those clones that are mutated are incorporated into the network.

2.3 Summary

The immune system can be considered to be a highly distributed and complex learning mechanism that is capable of recognising and learning new invaders and removing them from the system. (Jerne, 1974) proposed a network theory that attempted to describe how the immune system functions as a network. It is from this work that inspiration for work in (Timmis and Neal, 2000) is derived from. A resource limited artificial immune system was proposed, called AINE, which uses metaphors inspired by the immune network theory.

(Timmis and Neal, 2000) introduce the concept of an artificial recognition ball (ARB) object that describes the area in shape space that the data item can recognise a similar data item. AINE employs ARBs along with a resource allocation mechanism to limit the size of the resultant networks.

The following section deals with some issues found with AINE's resource allocation mechanism and presents modifications to correct these problems. Testing is then carried out on the modified algorithm using a simulated data set and the well known machine learning benchmark, the Iris data set (Fisher,1936).

3 Problem Identification and Revision

On re-implementation of AINE it was found, during testing, that the behaviour of the new version was unlike that seen in the original version. In the original version of AINE it was observed that over time the network size would fluctuate wildly, but would always be punctuated by periods of stability know as the core network. This core network size would remain constant in time, although may contain different ARBs each time, and formed the basis of the termination condition described in the previous section. However, during testing of the new version, it was found that there was an error in the algorithm that was causing the networks to degenerate to one ARB after only a few iterations. This was found to be caused by an error in the stimulation level calculation combined with a fault in the resource allocation mechanism.

The error in the resource allocation mechanism is identified and a revised version is proposed.

3.1 Stimulation Level of ARBs

In AINE, normalised ARB stimulation levels were being produced in the range of:

$$1.0 \geq stimulation < 0.0$$

where the normalised stimulation level was always greater than 0.0. This proved to be incorrect given that the stimulation levels for each ARB are calculated in the network, then normalised, which should produce stimulation levels in the range of:

$$1.0 \geq stimulation \leq 0.0$$

where normalised stimulation for two of the ARBs must have a value of 1.0 and 0.0 respectively. This produced networks that were degrading rapidly and no longer represented the data set to be learnt. It was then discovered that the degradation was caused by the resource allocation mechanism show in Figure 3.1. Resources are allocated to each ARB by the following function:

$$r_x = k.(sl)$$

(Timmis, 2000)

where r_x is the resources to allocate, k is a constant and sl is the stimulation level of the ARB. As the stimulation level for one ARB in the network is always going to be 0.0 (Due to normalisation) there would always be one ARB with zero resources.

```
Allocate B cells to ARB's based on stimulation level
Calculate number of B cells over allocated
if too many B cells have been allocated
      repeat
            get the weakest ARB
            if number of resources held by ARB is zero
                  Remove ARB from the network
            if the number of B cells to remove > number B cells
held by ARB
                  Remove ARB
                  Decrement number of B cells to remove by
number held by ARB
            else
                  Decrement ARB by number of B cells
            endif
      until number over allocated equals zero
else
      Go through network and remove all ARB with zero B cells
endif
```

Figure 3.1: Original resource allocation mechanism (Timmis, 2000a)

If an ARB has zero resources, then according to the resource allocation mechanism above, it would always be removed from the network.

It was evident that using the resource allocation method above, the weakest ARBs in the network were not getting a chance to adequately compete for resources. This means that the networks that were being produced would degrade to the point that there was only one ARB

left after a small number of iterations. The correction of this error is described in the next section.

3.2 The revision

To correct the behaviour of the revised version of AINE, ARBs with zero resources are no longer purged from the network. The resource allocation mechanism now only removes weak ARBs when the number of resources allocated to the network is greater the number of resources set by the user. This is achieved by using the algorithm described in figure 3.2 below.

```
Allocate B cells to ABR based on stimulation level
Calculate number of B cells over allocated
If too many B cells have been allocated
repeat
      Get the weakest ARB
      If number of B cells to remove > number B cells held by
ARB
            Remove ARB
            Decrement number of B cells to remove by number held
by ARB
      Else
            Decrement ARB by number of B cells
      Endif
Until number over allocated equals zero
```

Figure 3.2: New resource allocation mechanism.

This change in the resource allocation mechanism immediately had the desired affect and the revised AINE started to produce similar networks to the previous version. However an important side affect of the corrected algorithm appeared. The revised AINE, although exhibiting similar behaviour to AINE, is producing networks that slowly tend towards the strongest component of a given data set.

3.2.1 Revised process of execution for AINE.

The process of execution for AINE involves the integration of the mutated ARBs only if the termination condition has not been meet. This can however be revised given the following argument:

That mutated ARBs are useful to the core population because they provide some level of diversity based on a stochastic process.
The core population is the combination of both existing ARBs from the initial data set and new mutant clones produced by interactions with the training set.

Therefore a true representation of the core population is after the mutants have been incorporated into core population and the resource allocation mechanism has been executed. Given this argument the process of execution can be altered from that seen in figure 1.1 to the one shown in figure 3.3

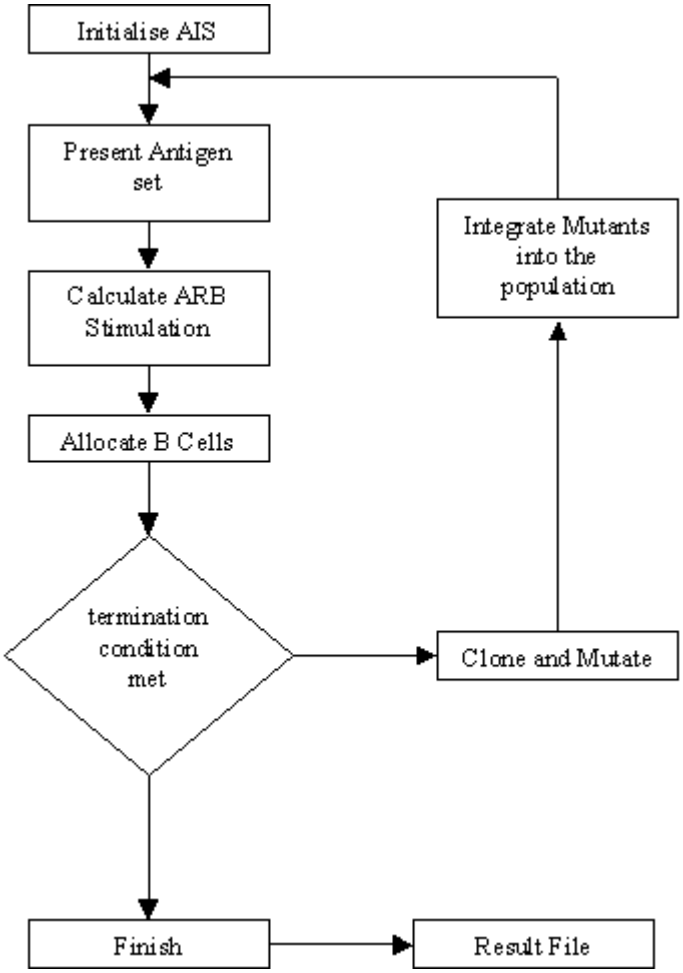


Figure 3.3: The revised process of execution

3.3 Summary

Having identified the need for correcting the known errors in AINE, a modified algorithm was developed. AINE still retains much of the ideas and implementation from the original version, but differences in execution and the resource-limited mechanism make its behaviour new. An error in the stimulation level calculation led to the new version exhibiting very rapid degradation of the networks. This was not the behaviour that was expected, the cause of this though was found to be the resource allocation mechanism, which has been rewritten to correct this behaviour.

The following section identifies suitable test methodologies and introduces the data sets that are tested on AINE.

4 Test Methodologies

In order to draw comparisons with the work of (Timmis, 2000b) the revised version of AINE will be tested using the same two data sets. The two original data sets are the *Simulated Data Set*, the *Fisher* Iris (Fisher, 1936) data set. To provide some level of coherence with the previous work the tests described in (Timmis, 2001b) will be reproduced to allow comparison of the new version against the previous version. This will also give some direction as to sensible parameter choices for the algorithm. The testing described in (Timmis, 2001) shows the affects of changing the following:

- The **Network Affinity Threshold** Scalar.
- The number of **Resources Allocated** to a network
- The **Mutation Rate**

The same test scheme will be applied to the revised version and the results analysed by looking at the changes in the network connectivity, network size and visualisation of the networks.
This section gives an introduction to the techniques employed to visualise and assess the performance of these networks. The data sets will be introduced and an explanation of the test strategy will be given for each set.

4.1 Network Visualisation

The networks are visualised using aiVis (Timmis, 2001) shown in (Figure 4.1). aiVis applies a simple attraction-repulsion algorithm to layout the network.

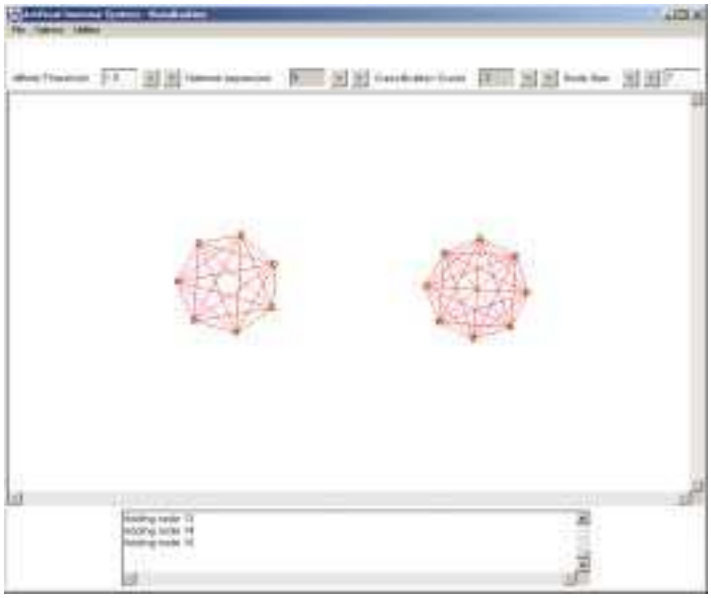


Figure 4.1 The initial Population of the Simulated Data Set visualised in aiVis.

Each link in the AIN (Artificial Immune network) represents the affinity between two ARBs. It should be stressed that the representations of the generated AIN are not meaningful as a conventional two-dimensional plot; they represent the information in the AIN as a topological structure. The aiVis tool also enables the user to present an unseen data item to the networks for classification; any nodes within the network that are closely matched to the unseen data item are connected by a link and the user can then make a visual interpretation of the unseen item.

4.2 Network Performance

It is possible to monitor the networks produced using AINE by running a separate test on the core population after each iteration. This is done by presenting ARB cells that are representative of an average data item for each known cluster to the core population. By binding these cells to every ARB in the core population and recording the number of links this process generates, it is possible to identify how well the networks represents the data set.

This is valid because an average data item will always produce a link to another ARB cell if it is of the same type. The results of these tests are then plotted and assessed.

4.3 Data Sets Used

Two data sets will be used to test the revised version of AINE, these are a simulated data set consisting of two linearly separable clusters and the Fisher Iris consisting of one linearly separable cluster and two in-separable clusters. Justifications for resource allocation, mutation rate and NAT settings will be given.

4.3.1 The Simulated Data Set

The simulated data set contains two linearly separable clusters of two-dimensional data (Figure 4.2) and in tabular form in Table 1. The data set is normalised before being passed into the AINE.

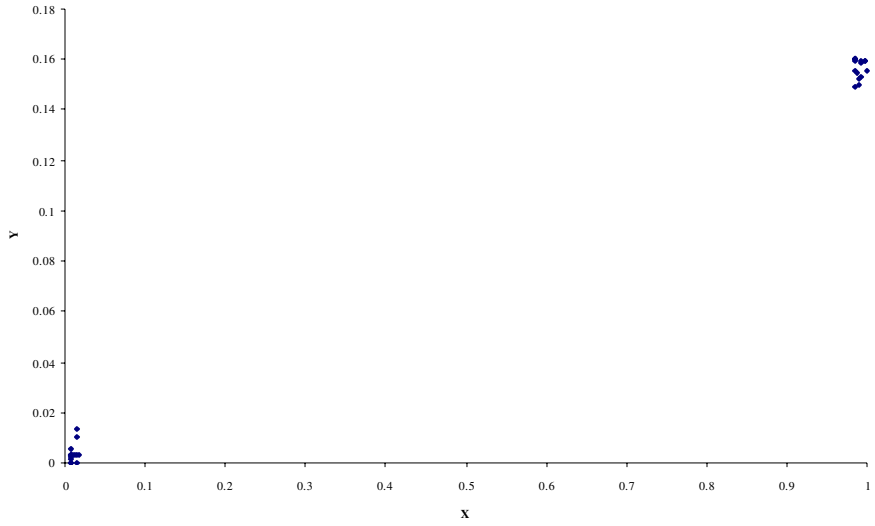


Figure 4.2: The Simulated Data Set in normalised form.

The Simulated Data set is divided into two parts, each representing half of each cluster, the first of these is used as the *Initial Population* and the second is the *Training Population*.

0.00708	0.00212
0.00779	0.00354
0.00779	0.00354
0.01487	0.00354
0.01629	0.00354
0.00850	0.00567
0.01416	0.00000
0.00779	0.00142

Table 1: The simulated data set in normalised form

4.3.1.1 Resource allocation for the simulated data set

The minimum number of resources that should be allocated to the simulated data network can be determined by running the network for one iteration over a number of different resource levels. This produces the graph shown in figure 4.3 and from this graph it is possible to determine that networks with resources to allocate below 90 for the simulated data set will not produce accurate representations of the data to be learnt. This is because the core population size is less than the initial population size. AINE will be tested with resource levels between the ranges of 90 and 200 resources. The value 90 is chosen because it is at this point there are enough resources present in the system for the network to initialise without the loss of the original data items.

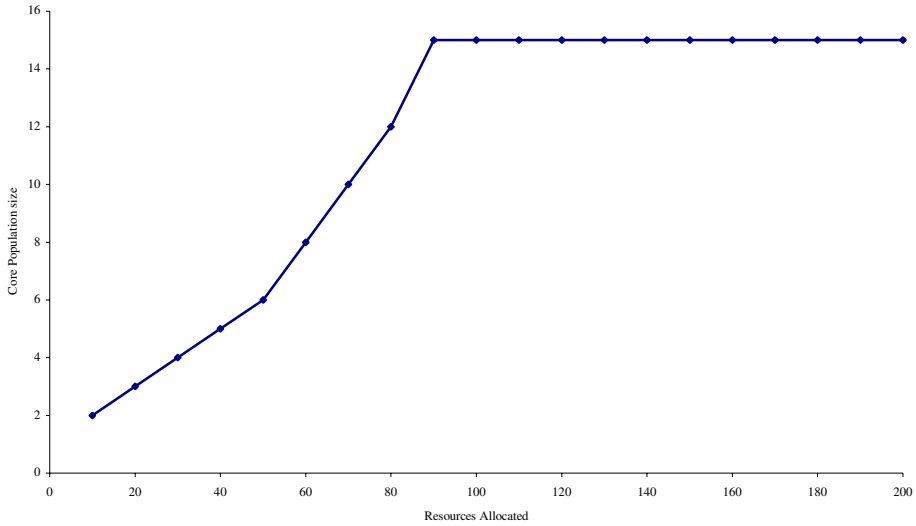


Figure 4.3 Showing the change in initial core population size given different maximum resource levels for the simulated data set.

4.3.1.2 Network Affinity Threshold's for the simulated data set

In the original version AINE the Network Affinity Threshold (NAT) controlled the connectivity of the networks and could potentially be used to reduce the complexity of the data set. Tests were run on NATs between 0.1 and 1.0 and the connectivity and size of the networks were examined. These test were designed to mimic those undertaken in previous work (Timmis, 2001).

4.3.1.3 Mutation Rates for the simulated data set

The Mutation Rate parameter used in AINE was found to affect the network connectivity or the network size. It is however an important controlling factor of how diverse the network is allowed to become. The Mutation Rate will again be tested and results for network connectivity and size presented in order to make a suitable comparison between the original

version of AINE and the revised version. Mutation Rates between 0.1 and 0.9 intervals of 0.1 were used to cover the entire range of possibilities.

4.3.2 The Iris Data Set

The Iris data set (Fisher, 1936) is a well-known machine learning benchmark. The data set contains three classes each of which represent a type of iris plant. The three types are *virginica*, *versicolor* and *setosa* and there are 50 instances of each. The data has four dimensions, petal length, petal width, sepal length and sepal width (Table 2).

Sepal Length	Sepal Width	Petal Length	Petal Width	Class
5.1	3.5	1.4	0.2	*Iris-setosa*
4.9	3	1.4	0.2	*Iris-setosa*
4.7	3.2	1.3	0.2	*Iris-setosa*
…	…	…	…	…
7	3.2	4.7	1.4	*Iris-versicolor*
6.4	3.2	4.5	1.5	*Iris-versicolor*
6.9	3.1	4.9	1.5	*Iris-versicolor*
…	…	…	…	…
6.3	3.3	6	2.5	*Iris-virginica*
5.8	2.7	5.1	1.9	*Iris-virginica*
7.1	3	5.9	2.1	*Iris-virginica*

Table 2: An example of the Iris data set

The setosa class is linearly separable from the other two classes, but the virginica and versicolor classes are linearly in-separable.

4.3.2.1 Resource allocation for the Iris data set

As for the simulated data set, there is a threshold value below which the initial network will not be representative of the data set to be learnt. This threshold is calculated by running AINE for one iteration over many different resource allocations. The results for the iris data set are shown in figure 4.4.

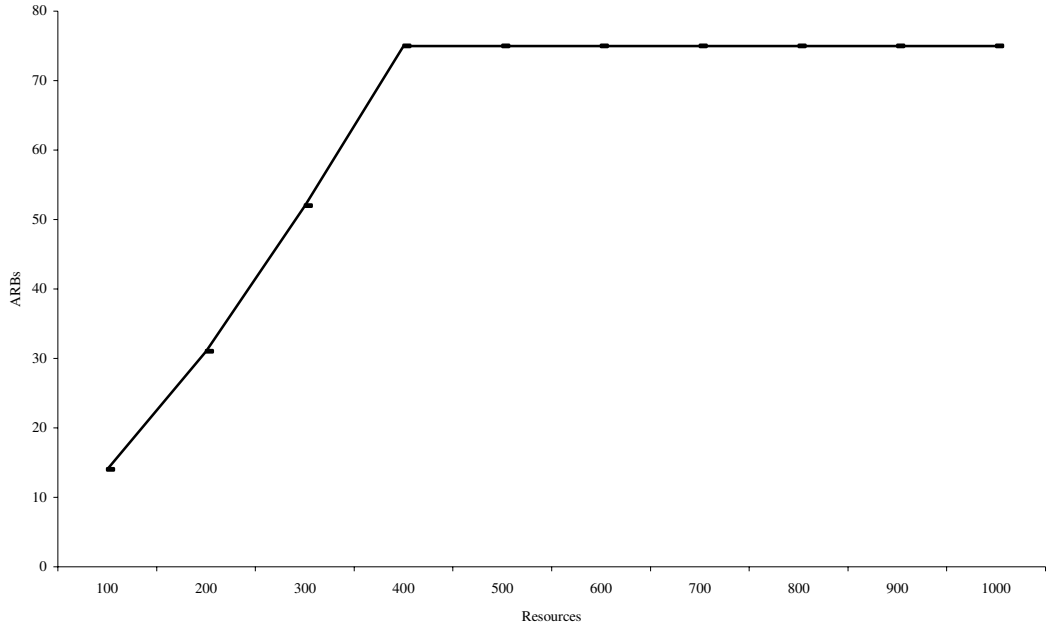


Figure 4.4 Showing the change in initial core population size given different maximum resource levels for the iris data set

As can be seen, it is only when the number of resources allocated exceed 400 that the core population is the same size as the initial population.

4.3.2.2 NAT's and Mutation Rates for the Iris data set

As for the simulated data set the NAT scalar and Mutation Rate will be tested through the full range of values, 0.0 – 1.0, at increments of 0.1 to add continuity between this work and that performed in (Timmis, 2001).

5 Results and Analysis

This section presents the results of the testing undertaken on AINE. Before presenting the results a summary of the previous test results from AINE are presented. The test results from the simulated data set and the Iris data set are then described and comparisons to the original version of AINE are drawn.

5.1 Summary of results from AINE

An investigation into the affects of the NAT Scalar, Mutation rate and Resources Allocation was undertaken for AINE, with respect to network connectivity and network size (Timmis, 2001). A summary of the findings are given below.

The Network Affinity Threshold (NAT) Scalar was incorporated into AIN to provide some control over the connectivity of the resulting network structures. It allows the user to define a threshold by which it can be said that two ARBs are connected (similar) in data-space. The test undertaken demonstrated that the NAT Scalar did indeed have a significant impact on network connectivity. Low NAT's (towards 0.0) produced networks that were more sparsely connected and Higher NAT's (towards 1.0) produced highly connected networks. The author

suggested that this behaviour could potentially be used if data is dense and separation of clusters is difficult.

The number of resources is part of the r*esource allocation mechanism* (Section 2.2.1.3) and represents the maximum number of resources that the entire network can posses. Resources are allocated to ARBs based on their stimulation levels. If the total number of resources allocated to all the ARBs in the network is greater than the maximum number of resources allowed, then the weakest ARBs are purged from the network until the total number of resources claimed is equal to the maximum number allowed. The results from testing on AINE suggested that the number of resources is directly proportional to the network size and that it has little or no effect on the connectivity of the networks produced.

The affect of the Mutation Rate is the final parameter described in (Timmis, 2001). It defines the probability of a clone mutating and is used to produce some degree of diversity in the core population derived from highly stimulated ARBs. Results from AINE suggest that is has no real affect on network connectivity or network size.

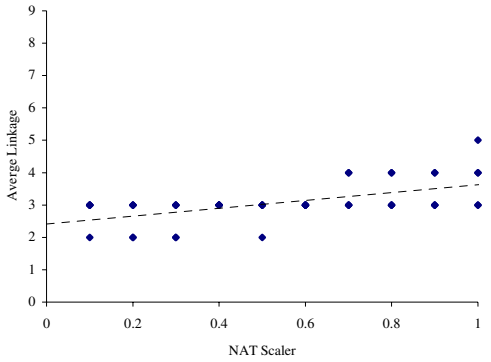
5.2 The Simulated Data Set

5.2.1 Changing the Network Affinity Threshold Scalar

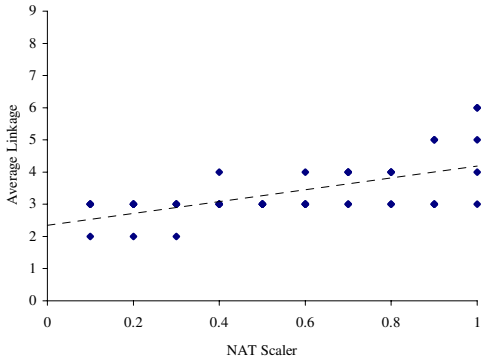
The Network Affinity Threshold (NAT) Scalar from previous experience with AINE should allow the user to define how well connected the resultant networks are. The NAT Scalar should have little or no effect on the network size. Samples of the networks were taken at iterations 2, 5 and 10 and the results displayed (see below). It is expected that AINE will display the same behaviour as AINE. The Resources Allocated to this test were 150 and a Mutation Rate of 0.4.

5.2.1.1 Network Connectivity

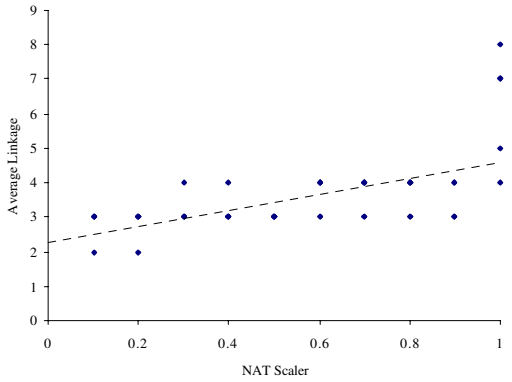
Figure 5.1 shows the affect of increasing the NAT scalar on the network connectivity. The average numbers of links were plotted for five different runs, for a range of possible values. Measurements were taken at three different time intervals (2, 5, and 10 iterations) to show that the affect is not a localised effect. It can be seen that with an increasing NAT Scalar, the connectivity of the networks produces also increases. This affect is similar to the affect seen in AINE and is what was expected to be seen.



(a) Network after 2 iterations shows a positive relationship between NAT Scalar and network connectivity



(b) Network after 5 iterations showing an increased positive relationship between NAT Scalar and network connectivity

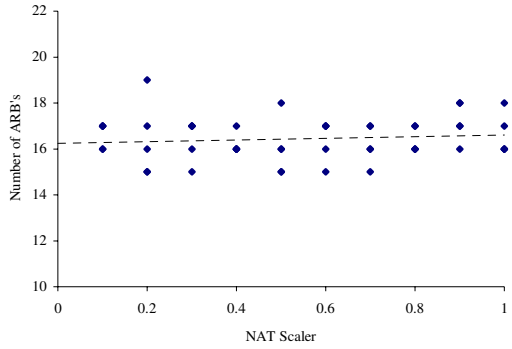


(c) After 10 iterations there is a significant positive relationship between NAT Scalar and network connectivity

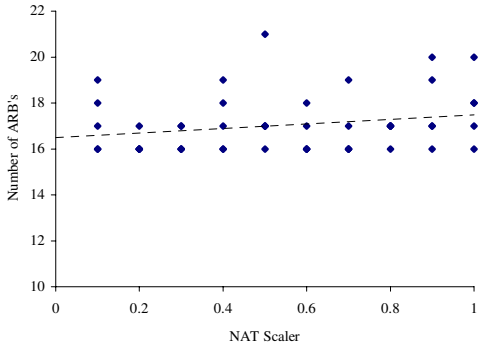
Figure 5.1 The affect of the NAT Scalar on network connectivity

5.2.1.2 Network Size

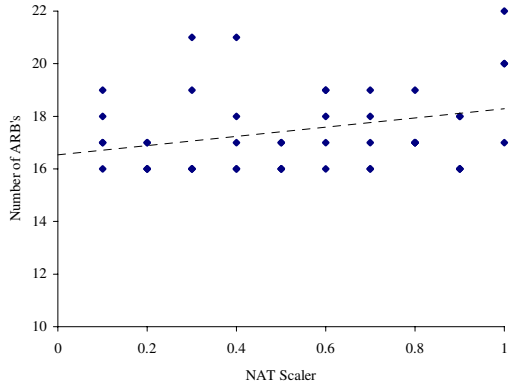
Figure 5.2 shows the affect of the NAT Scalar on the network size. Again the plots represent the average results of five runs plotted over a range of NAT Scalars. Three plots at different time intervals are shown to guard against the possibility of localised effects. It can be seen that at iteration 2, the network size is stable across that range of NAT Scalars, and this was what was expected. However, iterations 5 and 10 start to show an increasing gradient in the average size that does not follow the expected pattern. However this is not believed to be significant and can be attributed to the algorithm tending towards optimisation and not stability, and is discussed later in this section. On further study of the plots for iterations 5 and 10 it can be seen that there are a number of high network sizes that are forcing the average towards a stronger gradient, but the underlying base level remains the same. Suggesting that the NAT Scalar does not have a significant affect on the network size and the effects seen in the plot are due to other behavioural factors in the algorithm.



(a) Network after 2 iterations showing slight positive relationship between NAT Scalar and network size



(b) Network after 5 iterations showing an increased positive relationship between NAT Scalar and network size



(c) After 10 iterations there is a very marked positive relationship between NAT Scalar and network size

Figure 5.2 The affect of the NAT Scalar on network size

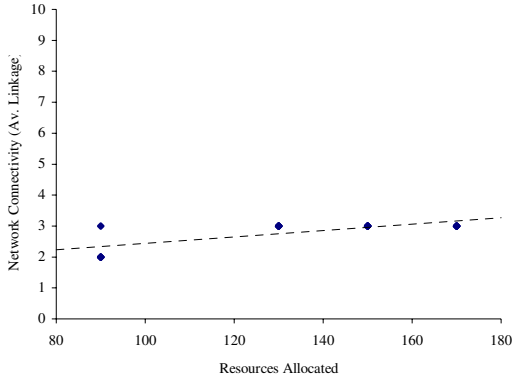
5.2.1.3 Summary of Results

The results show that the NAT Scalar is an effective control mechanism for reducing or increasing the connectivity of the networks produced. It is believed to have little effect on the network sizes. This suggests that one possible use for the NAT Scalar is as a mechanism by which the complexity of data can be reduced to identify key features in the data space. The results reaffirm those described in (Timmis, 2001).

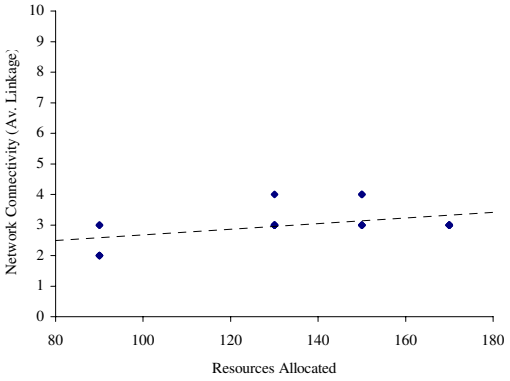
5.2.2 Changing the Number of Resources

Changing the number of resources allocated to the network is significant in limiting the size and growth of the networks. The number of resources defines the maximum number of resources that the network can claim on any one iteration and provides a limiting mechanism by which ARBs are removed if the total resources claimed are greater than the number of resources to allocate.

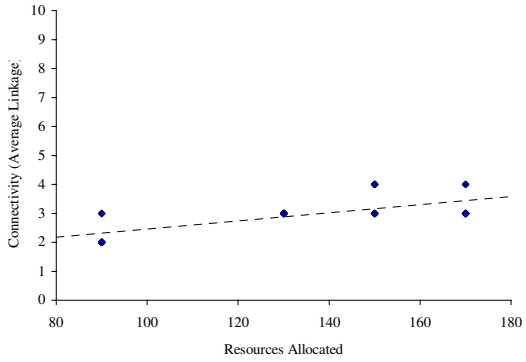
5.2.2.1 Network Connectivity



(a) Network after 2 iterations showing a positive correlation between the number of resources allocated and the connectivity of the network



(b) Network after 5 iterations showing a positive correlation between the number of resources allocated and the connectivity of the network

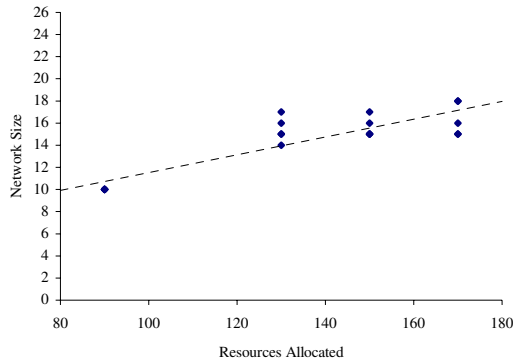


(c) Network after 10 iterations showing a positive correlation between the number of resources allocated and the connectivity of the network

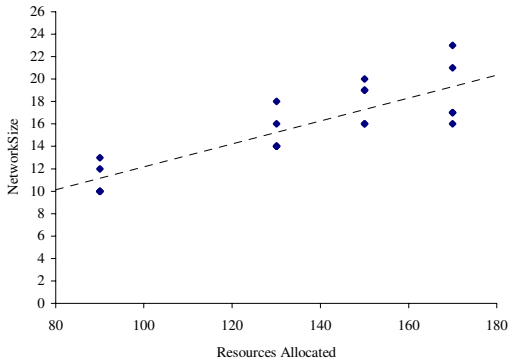
Figure 5.3 The affect of Resources on the network connectivity

Figure 5.3 shows that affect of changing resources levels on the network connectivity. Each plot represents five runs where measurements were made at 90 130 150 and 170 resources over three periods of time. The Trend-line represents the average the five runs. It was expected that the network connectivity would remain stable through the changing resources but it would appear that there is a slight positive gradient (about 0.015x). This is however not significant and can be attributed to the potential for larger network sizes at higher resource levels where the networks are not as sparse as at low resource levels.

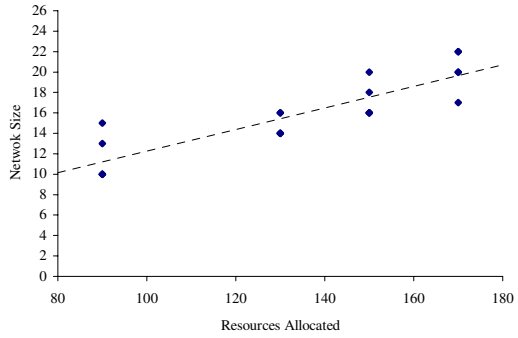
5.2.2.2 Network Size



(a) Network after 2 iterations showing a positive correlation between the number of resources allocated and the size of the network



(a) Network after 5 iterations showing a positive correlation between the number of resources allocated and the size of the network



(a) Network after 2 iterations showing a positive correlation between the number of resources allocated and the size of the network

Figure 5.4 The affect of changing resources on the network size

Figure 5.4 shows the changing network size over a range of resource levels. Again three time intervals were chosen to ensure consistency. The result show that as the number of resources available to the system increase so does the network size. This is as expected because as more resources become available to the network, weaker ARBs are given a greater chance to survive and proliferate.

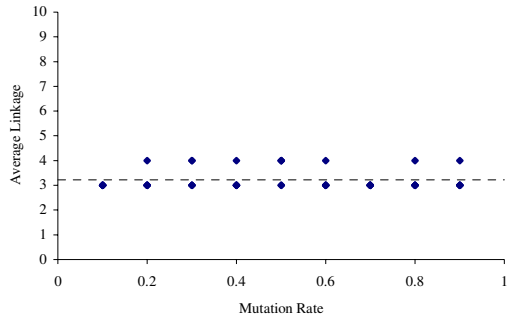
5.2.2.3 Summary

Results show that the number of resources allocated to AINE controls the size of the networks produced. It also suggests that to allow weaker ARBs to survive longer, a higher number of resources are required. The number of resources does not appear to have a significant affect on the network connectivity. These results confirm those presented in (Timmis, 2001).

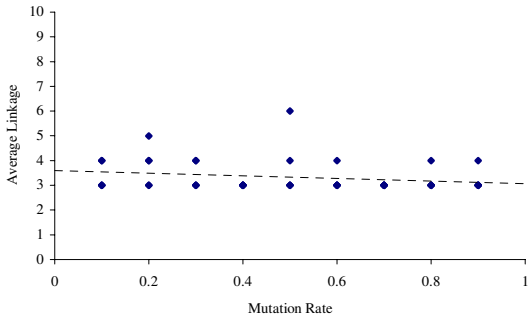
5.2.3 Changing the Mutation Rate

The Mutation Rate is a threshold value that determines whether a cloned ARB has one or more of its data values mutated. Results from AINE showed that the mutation rate had little affect on either the network connectivity or the network size. The results from AINE can be seen below.

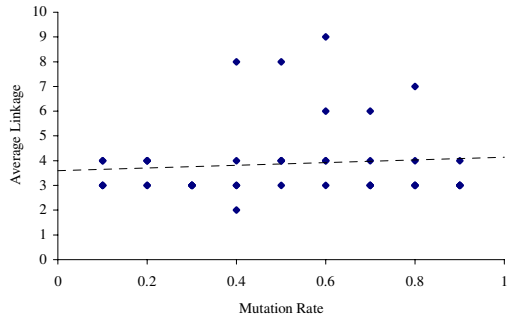
5.2.3.1 Network Connectivity



(a) Network after 2 iterations showing no correlation between the mutation rate and the network connectivity.



(b) Network after 5 iterations showing a slight negative correlation between mutation rate and network connectivity.

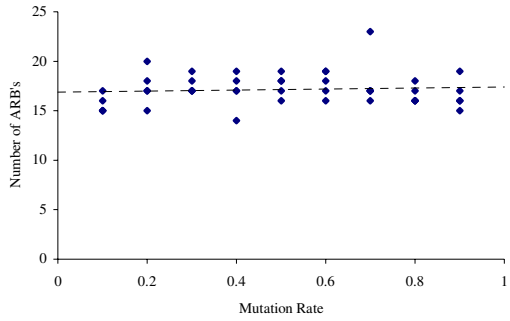


(c) Network after 10 iterations showing slight positive correlation between mutation rate and network connectivity.

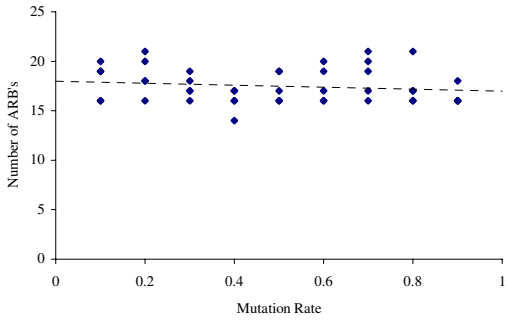
Figure 5.5 The affect of changing mutation rate on network connectivity

Figure 5.5 shows the affect of changing the mutation rate on network connectivity. It can be seen that the mutation rate has little affect on the connectivity over the range 0.1 – 0.9. At iteration 2 the pattern is very stable with an almost horizontal line, but as the number of iteration increases the pattern becomes less stable. The shift in the pattern is both negative (iteration 5) and positive (iteration 10) and therefore suggests that these are only slight fluctuations. The results are as expected and are similar to those seen in earlier work.

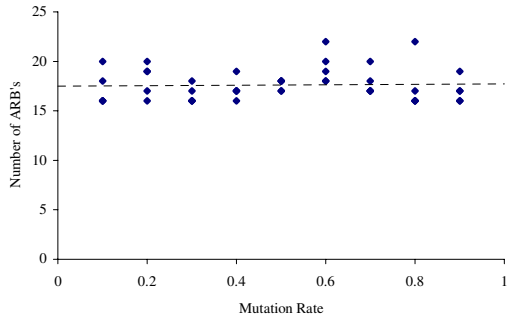
5.2.3.2 Network Size



(a) Network after 2 iterations showing no distinct correlation between mutation rate and network size.



(b) Network after 5 iterations showing a slight negative correlation between mutation rate and network size.



(c) Network after 10 iterations again showing no correlation between mutation rate and network size

Figure 5.6 The affect of changing the mutation rate on network size

Figure 5.6 shows the affect of changing the mutation rate on network size. The results show that the mutation rate has no effect on the network size. This again confirms the results from AINE.

5.2.3.3 Summary

The Mutation Rate can be said to have little effect on both network connectivity and size. This was as expected because the mutation rate is a parameter that is designed to alter the diversity of the networks, not the size or connectivity.

5.2.4 Summary of results from the Simulated Data set

AINE was tested on a simulated data set that was linearly separable using the same conditions and parameters that were used to test AINE. The goal was to compare the performance of AINE against AINE.
In comparison AINE produced similar results to AINE in all tests and the following things can be concluded.

- Changing the NAT Scalar is an effective mechanism for increasing/decreasing the connectivity of the networks produced

- Changing the resources allocated to each network also increases/decreases the network connectivity, but also controls the network size.
- Changing the Mutation Rate has no effect on either connectivity or network size but affects the diversity of the networks produced (described in the next section).

5.2.5 Network Evolution

ANIE was run with the maximum number of resources-to-allocate set at 90 and networks were produced over 200 iterations. The changes in the core population size and the average linkage can be seen in figure 5.7. It can be seen that the network remains stable for the first three iterations were the network size is the same as the initial population. Then on the 4th iteration the network size plummets to 10 ARBs and remains at this level until the 60th iteration. Between the 60th and the 160th iteration the network undergoes periods of fluctuation and stability much like the human immune network is believed to experience. It should be noted that when the core population size is stable, the networks produced on each iteration are all exactly the same as each other.

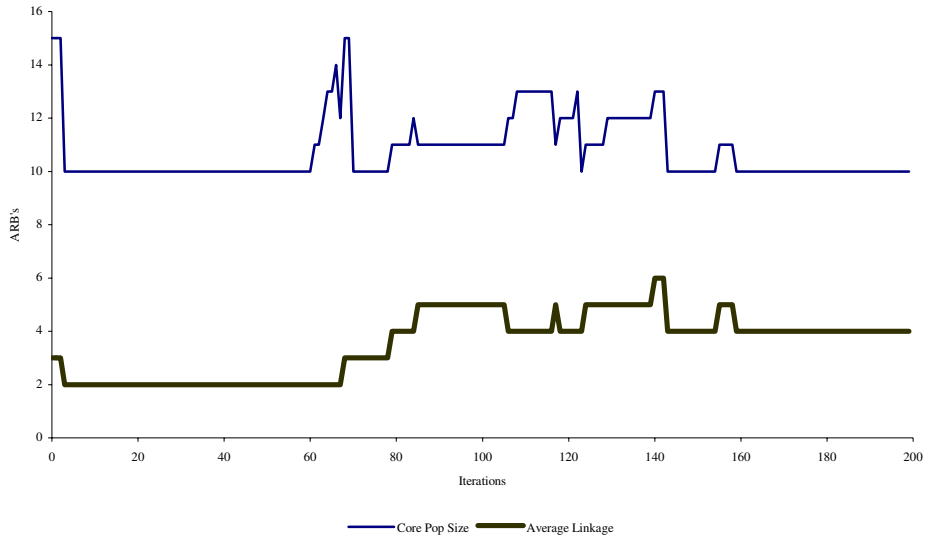
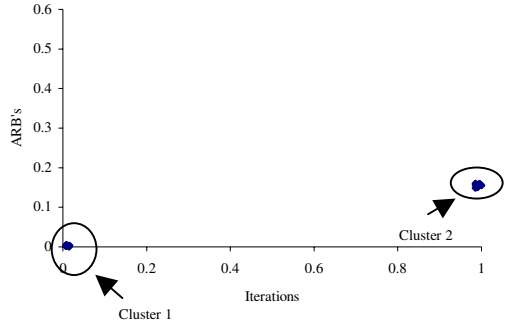
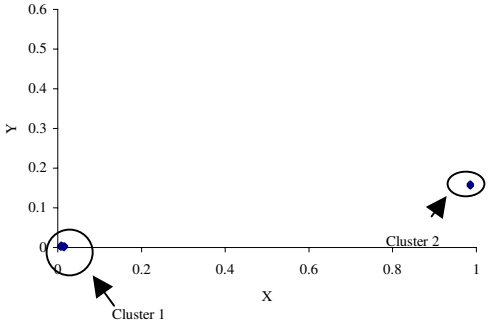


Figure 5.7 Core population levels and average linkage between iterations 0 – 200 with a Resource level of 90 and a Mutation rate of 0.1

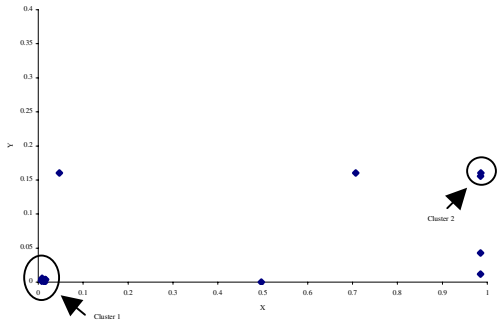
5.2.5.1 Evolving the simulated data set network



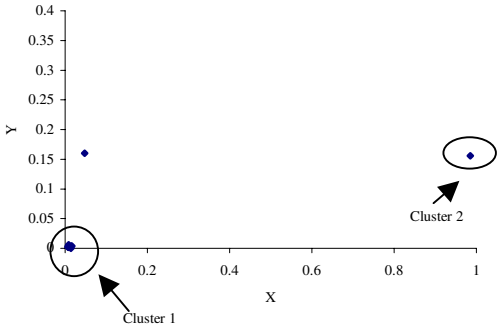
(a) Network after 2 iterations. The clusters are the same as the initial network



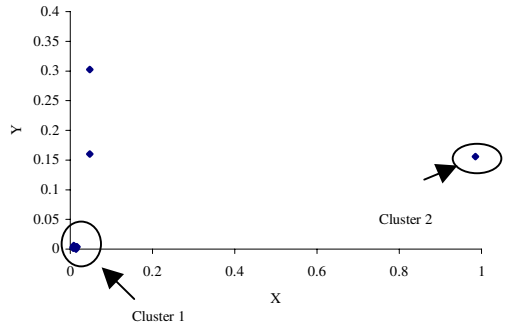
(b) Network after 10 iterations. The Clusters are still located in the same positions. Cluster 2 has lost a number of it data points.



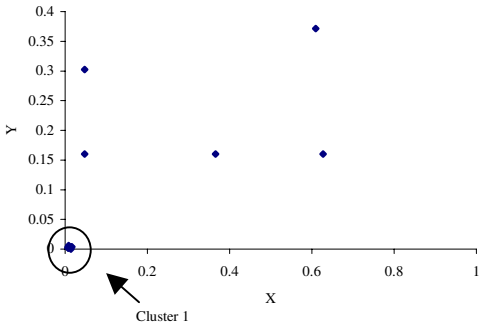
(c) Network after 68 iterations. The two clusters are still present and a number of new outliers are now visible.



(d) Network after 77 iterations. Cluster two in now only comprised of one data item.



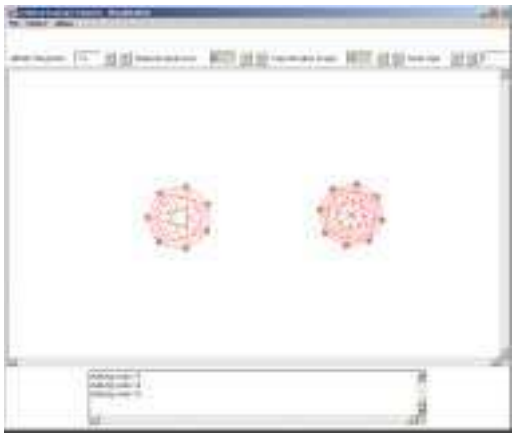
(e) Network after 80 iterations: Cluster 2 is still persisting in the network, but still only with one data item.



(f) Network after 113 iterations: Cluster two has been completely removed. Only Cluster one and several outliers remain.

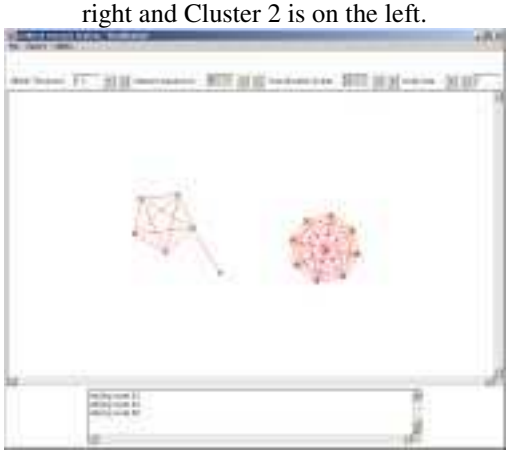
Figure 5.8(a-f): Two-dimensional plot showing the evolution of the simulated data set using AINE.

Figures 5.8 show the evolution of the network between 0 and 113 iterations. Each figure is taken at points of stability in the network structure. It can be seen that early networks represent the simulated data set quite closely, but as time progresses the networks start too include a certain number of random data points that are generated by the cloning and mutation operations of AINE. It can also be seen that the weaker data cluster (Cluster 2) slowly degrades over time, finally disappearing around iteration 113. The earlier network structures are very encouraging, but it is not until these are visualised is it possible to see if there are any meaningful structures in the network.

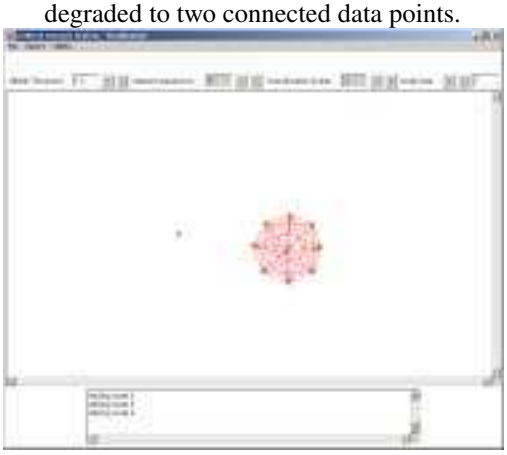


(a) Network after 2 iterations. Cluster 1 is on the right and Cluster 2 is on the left.

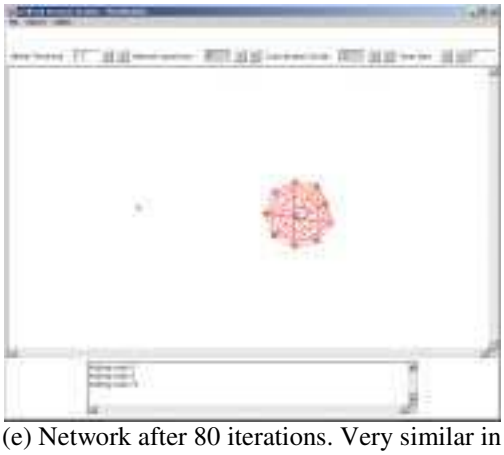
(b) Network after 10 iterations. Cluster 2 has degraded to two connected data points.



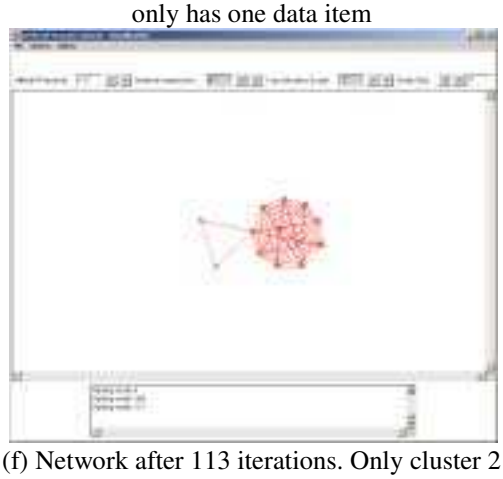
(c) Network after 68 iterations. Mutant data points are now visible in the network



(d) Network after 77 iterations. Cluster 2 now only has one data item



(e) Network after 80 iterations. Very similar in composition to iteration 77.



(f) Network after 113 iterations. Only cluster 2 remains, with attached mutant data points.

Figure 5.9 (a-f) Networks viewed using aiVis for the simulated data set

Figures 5.9 (a-f) shows the evolving networks as displayed in the aiVis tool. Networks 3.4a-c are very interesting because they show the transition of cluster 2 from the initial network, (a), to a reduced version, (b), and then to a new version that includes mutants that represent data items that are similar to the original data. It is this behaviour that appears to follow the theory of how the immune system works. It is important to note that once a representation of cluster 2 is lost from the network, it never returns, but this point in the network should not be reached and AINE will terminate once a stable pattern has been found.

5.2.5.2 Trends in the Simulated Data set

Two average ARB cells were calculated from the whole of the simulated data set. AINE was then tested with the following setting: NAT Scalar 0.4, Mutation Rate 0.1 and Resources 150. These values were chosen because on average these produced what could be considered as good networks. The two test cells were then tested against the core populations produced by every iteration of the test run. The results obtained for this test were plotted in figure 10.

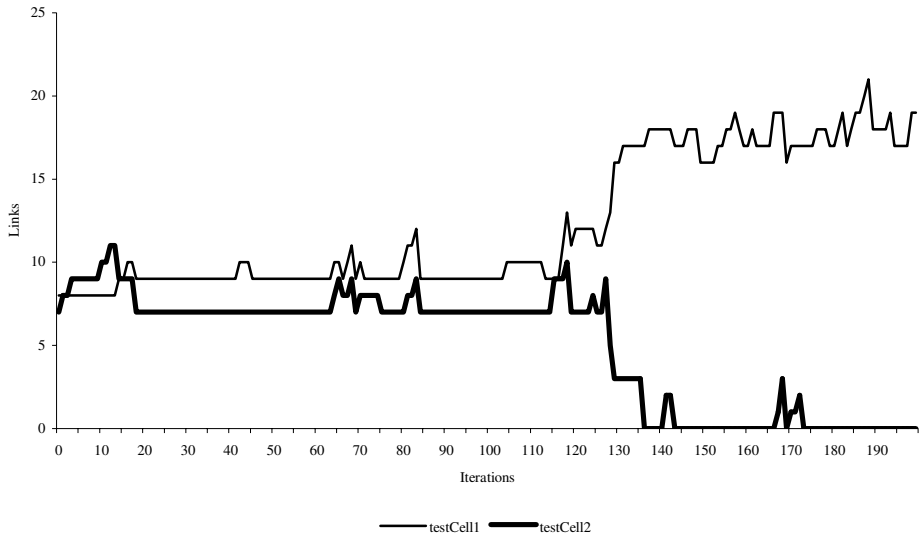
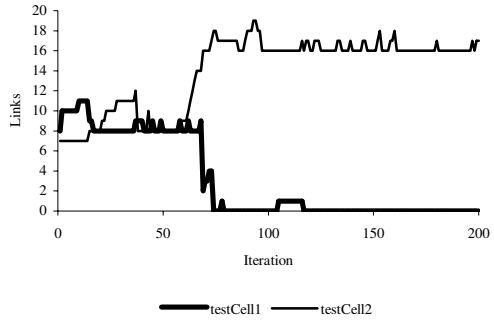
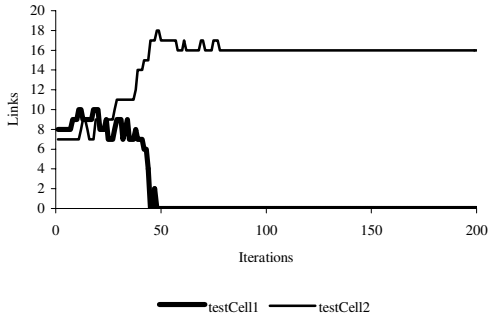


Figure 5.10 Trends showing degradation of test cell 2 over time.

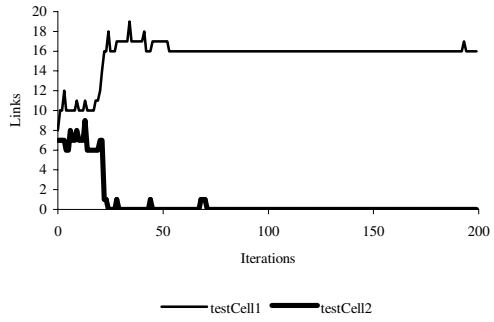
Figure 5.10 shows the slow degradation of one of the clusters from the simulated data set. This is an example of the AINE tending towards the strongest cluster in a data set. The results are the same for varying resources levels. This clearly shows the elitist strategy that AINE now possess, where after 120 iterations the number of cells test cell 2 is linked to drops down to zero. Looking at the actual 2D plots of the data shows that as the cluster similar to test cell 2 dies, the other cluster grows in size. This point at which the networks start to deteriorate seems to be controlled by the Mutation Rate. Figures 5.11(a-d) shows trend plots for mutation rates at 0.1, 0.3, 0.7 and 0.9 respectively. It is clear from these plots that as the Mutation Rate is increased the period before degradation decreases. One possible cause of this is that as the Mutation Rate increases there exists a greater chance that a highly stimulated ARB will mutate. If the chance of mutating is higher, more ARBs will be mutated. As the most highly stimulated ARBs are more susceptible to mutation it follows that the strongest cluster will proliferate, as this group will produce more clones and the weaker cluster will die off.



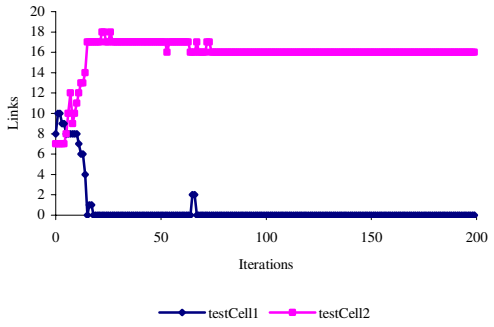
(a) Mutation Rate of 0.1. Network deteriorates after 65th iteration.



(b) Mutation Rate 0.3. Network deteriorates after 45th iteration.



(c) Mutation Rate 0.7. Network deteriorates after 20th iteration.



(d) Mutation Rate 0.9. Network deteriorates after 15th iteration.

Figure 5.11 Trends in the networks produced using the simulated data set using different mutation rates

5.2.5.3 Summary of Network Evolution

The networks that evolve in AINE are representations of the data set being learnt. These representations exist in the form of clusters and can be visualised using aiVis and 2D plots. On testing AINE using a simulated data set (that has two distinct clusters) it has been possible to determine the behaviour of AINE. The algorithm is capable of discovering the core pattern that exists in the data set after only a few iterations and can be identified by stable periods in the network. However, as the algorithm continues it begins to tend towards the strongest cluster in the data set, essentially trying to find the optimal cluster or most stimulated group of ARBs.

5.3 The Iris Data Set

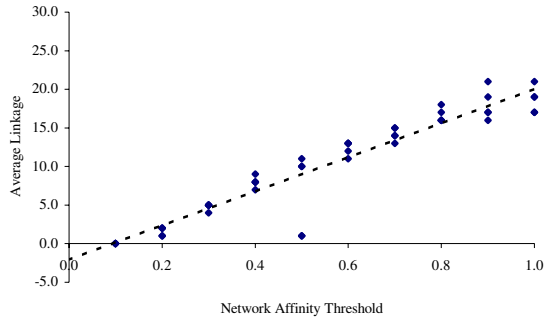
5.3.1 Changing the NAT Scalar

The results from the previous testing on the Iris data set showed the same results as for the simulated data set. The network connectivity showed a positive correlation with the increasing NAT Scalar, but there was no correlation between the NAT Scalar and the network

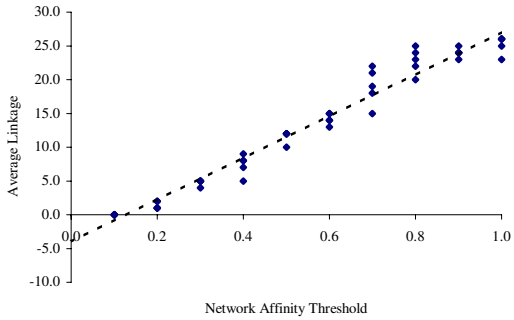
size. Shown below are the results for the iris data set testing the effect of the NAT Scalar on network connectivity and network size. The tests were repeated 5 times using a mutation rate of 0.1 and a max. number of resources of 450.

5.3.1.1 Network Connectivity

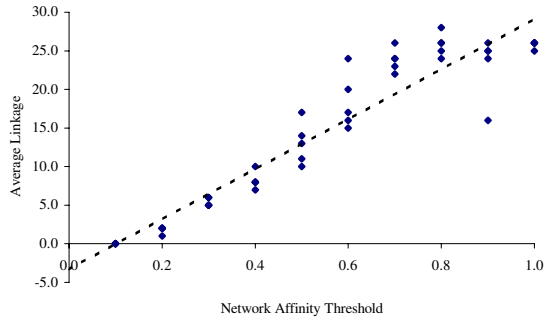
Figure 5.12 (a-c) show the affect of changing the NAT scalar on the connectivity of the networks produced by AINE. As in the tests on the simulated data set, 3 measurements were taken at iterations 2, 5 and 10, and each test was repeated 5 times. The three charts below all show strong positive correlations between the NAT Scalar and the connectivity of the networks produced. These results are similar to those from the simulated data set and match those from test on the previous algorithm.



(a) Network after 2 iterations clearly showing a positive correlation between NAT Scalar and network connectivity.



(b) Network after 5 iterations clearly showing a positive correlation between NAT Scalar and network connectivity.



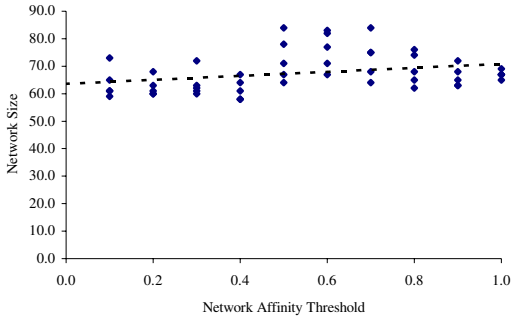
(c) Network after 10 iterations clearly showing a positive correlation between NAT Scalar and network connectivity.

Figure 5.12 (a-c): The affect of the NAT Scalar on the network connectivity of the iris data set.

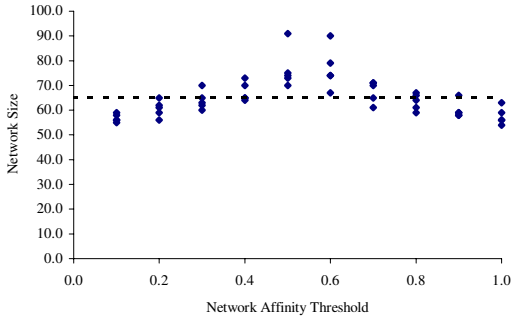
5.3.1.2 Network Size

Figure 5.13 (a-c) show the affect of the NAT Scalar on the size of the networks produced by AINE. Again the plots represent the average of five runs plotted over a range of NAT Scalars. Three plots at different time intervals are shown to remove the possibility of localised effects. At iteration 2 the plot shows a slight positive correlation between NAT Scalar and network size, but it is not a significant correlation. Iterations 5 and 10 both exhibit bell shaped curves, where the network size peaks at a NAT Scalar of about 0.5 This would appear to be a threshold value for this particular data set. As the NAT Scalar approaches 0.5 there is healthy competition between all three classes, but after 0.5, the Virginica and Versicolor classes become dominant and the Setosa class does not survive beyond the first few iterations. This

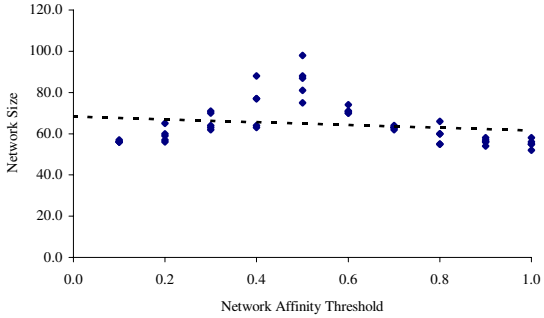
means that the two surviving classes claim more resources per data item and therefore the number of actual data points in the system drops because of the resource allocation mechanism.



(a) Network after 2 iterations showing a slight positive correlation between the NAT Scalar and the network size.



(b) Network after 5 iterations showing no correlation between the NAT Scalar and the network size, but showing an interesting bell shaped curve.



(c) Network after 5 iterations showing a slight negative correlation between the NAT Scalar and the network size, but showing an interesting bell shaped curve.

Figure 5.13 (a-c): The affect of the NAT Scalar on network size of the iris data set.

5.3.1.3 Summary

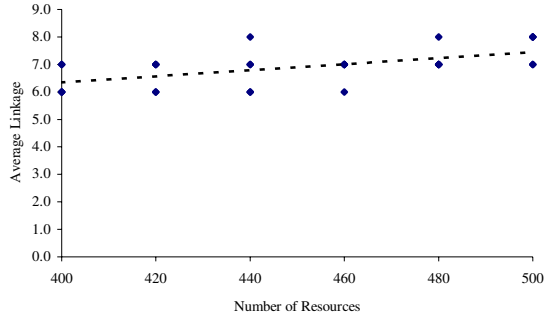
The results show that the NAT scalar behaves as it was expected to but has introduced the concept that in certain conditions there can be a threshold value for a parameter, beyond which the behaviour is not predictable. However it is felt that this could also be a result of some other influence and requires further testing and evaluation.

5.3.2 Changing the Number of Resources

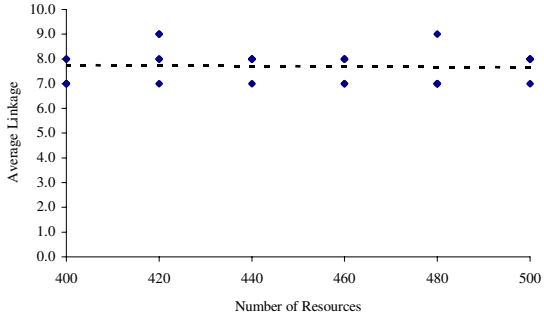
The results from the previous testing on the Iris data set showed the same results as for the simulated data set. The network size showed a positive correlation with the increasing NAT Scalar, but there was no correlation between the NAT Scalar and the network connectivity. Shown below are the results for the iris data set testing the effect of the NAT Scalar on network connectivity.

5.3.2.1 Network Connectivity

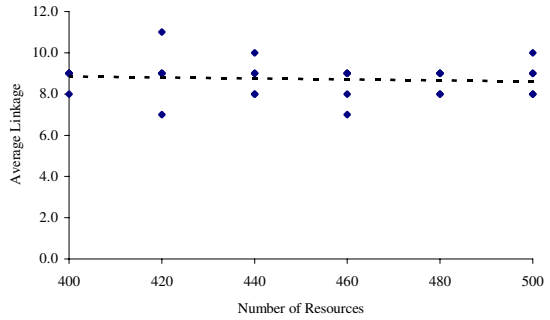
Figure 5.14 (a-c) again shows the results of five separate runs at three different time intervals. In 5.14a it can be seen that there is slight positive correlation between the number of resources and the network connectivity, but this is not significant. Figures 5.14 a & b both show a slight negative correlation which is again not significant. As a result it can be said that altering the number of resources available to the system does not affect the connectivity.



(a) Network after 2 iterations showing a slight positive correlation between the number of resources and the network connectivity.



(b) Network after 5 iterations showing no correlation between the number of resources and the network connectivity.

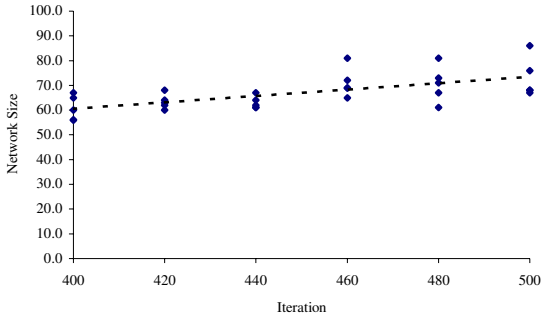


(c) Network after 10 iterations showing a slight negative correlation between the number of resources and the network connectivity.

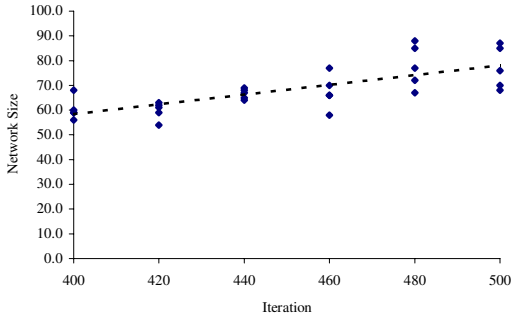
Figure 5.14 (a-c): The affect of changing the number of resources on network connectivity

5.3.2.2 Network Size

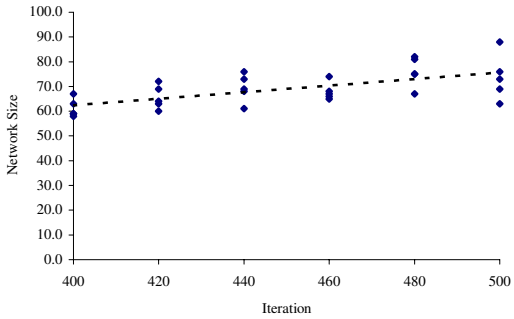
Figure 5.15 (a-c) represents the relationship between network size and the number of resources allocated to the system. It is clear to see that for all three time-intervals there is a positive correlation between the number of resources and the network size. This is the expected result as the number of resources limits the size of the networks and should control the population size.



(a) Network after 2 iterations showing a positive correlation between the number of resources and network size.



(b) Network after 5 iterations showing a positive correlation between the number of resources and the network size.



(c) Network after 10 iterations showing a positive correlation between the number of resources and the network size.

Figure 5.15 (a-c): The affect of the number of resources on network size of the iris data set.

5.3.2.3 Summary

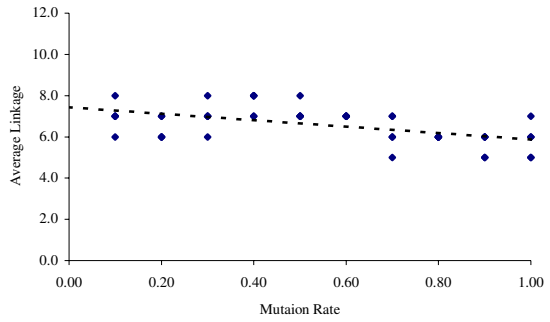
The results clearly show that altering the number of resources available to the system allows control over the maximum size of the networks and yet has no effect on the connectivity. These results complement those in (Timmis, 2001) and those seen for the simulated data set.

5.3.3 Changing the Mutation Rate

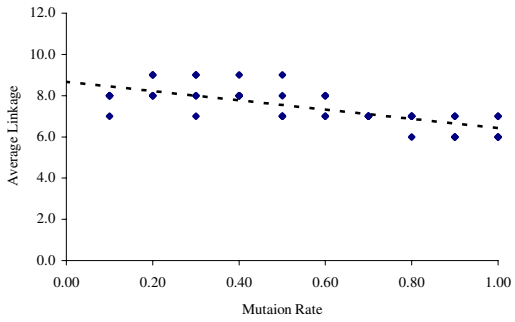
The results from the previous testing on the Iris data set showed the same results as for the simulated data set. Both the network size and the network connectivity are not affect by changing the mutation rate. The mutation rate allows control over the diversity of the resulting populations, where a higher mutation rate will lead to a more diverse population.

5.3.3.1 Network Connectivity

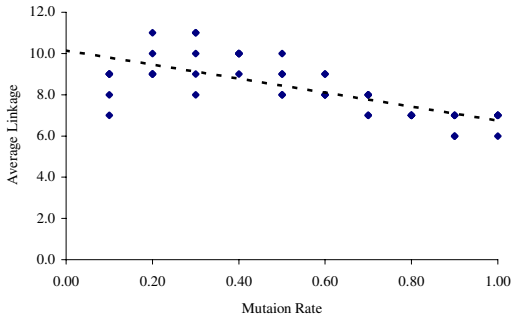
Figures 5.16(a-c) show the affect of changing the mutation rate on network connectivity for the iris data set. All three time-intervals show a negative correlation between mutation rate and connectivity, however none of these is very significant although it possibly suggests that there is chance that as the mutation rate nears 1.0 the networks are more diverse and this could reduce the connectivity of the networks slightly.



(a) Network after 2 iterations showing a slight negative correlation between the mutation rate and the network connectivity.



(b) Network after 5 iterations again showing a slight negative correlation between the mutation rate and the network connectivity.

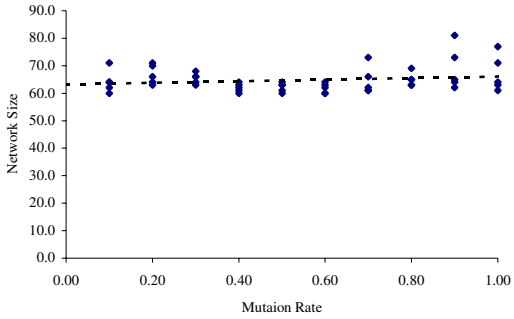


(c) Network after 10 iterations showing a more significant negative correlation between the mutation rate and network connectivity.

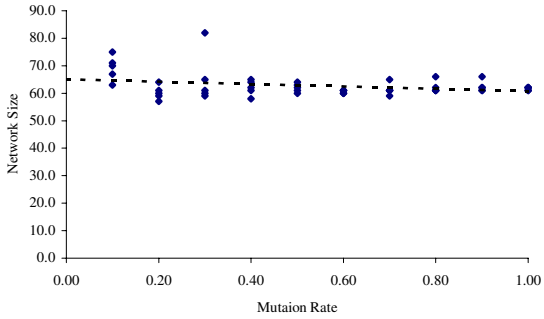
Figure 5.16 (a-c): The affect of the mutation rate on the network connectivity of the iris data set.

5.3.3.2 Network Size

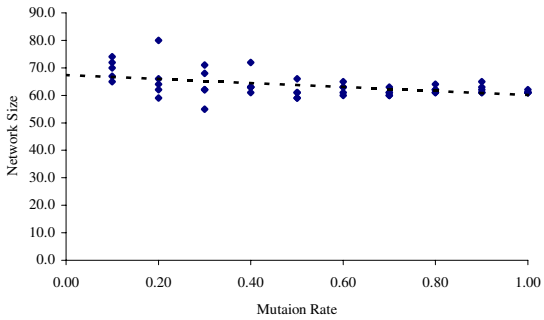
Figures 5.17 (a-c) again show the three plots at different iterations in the learning process, but this time show how the mutation rate affect the size of the networks. It can be seen that for all three plots there is no discernable correlation. This is expected as the mutation rate allows control over the diversity of the network, and not the size.



(a) Network after 2 iterations showing a slight positive correlation between the size of the network and the mutation rate.



(b) Network after 5 iterations showing a slight negative correlation between the mutation rate and the network size.



(c) Network after 10 iterations again showing a slight negative correlation between the mutation rate and the network size.

Figure 5.17 (a-c): The average affect of changing the mutation rate on the resultant network size at three different iterations

5.3.3.3 Summary

It has been shown that altering the mutation rate may have a slight affect on the network connectivity but this is not a significant, and the mutation rate has no effect on the network size. These results are similar to those obtained from the simulated data set and compare well to those see in (Timmis, 2001). The affect of the mutation rate on diversity needs to be addressed, but

5.3.4 Summary of the results from the Iris data set

The revised version AINE was tested on the Iris data set that contains three classes, two of which are not linearly separable. The goal was to compare the performance of the new version of AINE against the previous version. In comparison the revised version produced similar results apart from those produced by altering the NAT scalar. When the NAT Scalar was tested for the revised version a bell shaped curve was observed in network sizes (Figures 5.13 (a-c)). This behaviour has been attributed to a threshold that is determined by the competition between the three classes, were the peak represents the point at which all three classes are represented in the network, and all points after that only the strongest survive. These conclusion are however speculative and require further testing that will not be presented in this report.

5.3.5 Network Evolution

The revised version of AINE was run with the maximum number of resources set at 450 a mutation rate of 0.4 and a NAT Scalar of 0.3 for 100 iterations. The changes in the core population size and the average linkage can be seen in figure 5.18. It can be seen that the core population is very unstable for long periods of time and is occasionally punctuated by periods of stability (two consecutive iterations where the core population remains stable). On closer inspection it can be seen that there are periods of stability at iterations 3-4, 10-11, 51-52 and 66-67. It is important to note that even though the periods of stability are not as long as those seen for the simulated data set they are still import indicators of stable patterns.

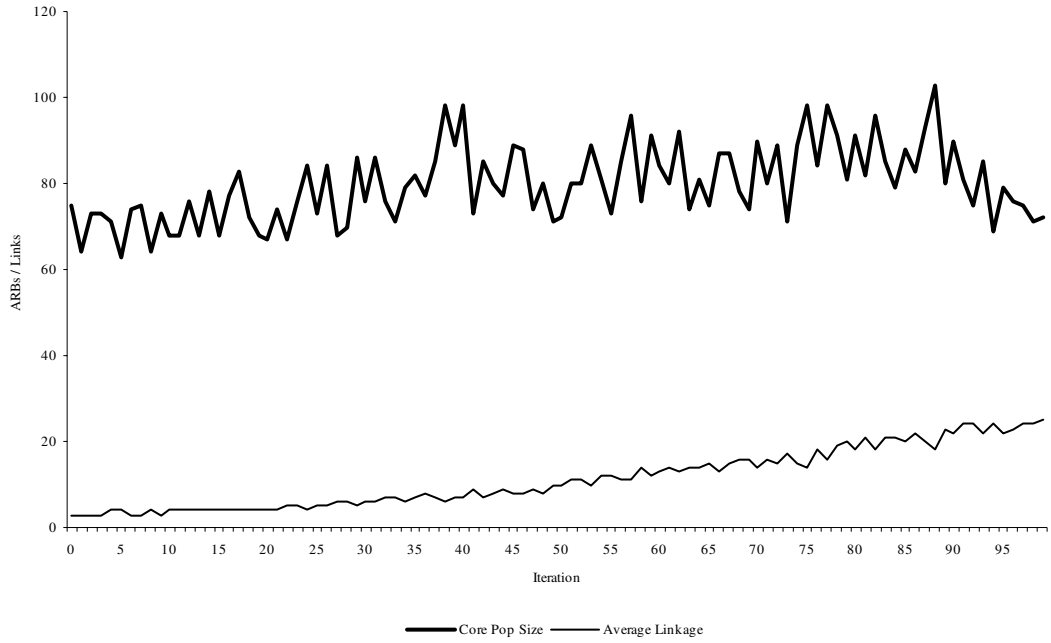
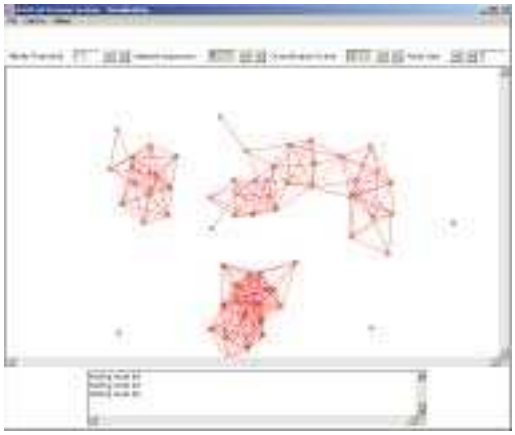
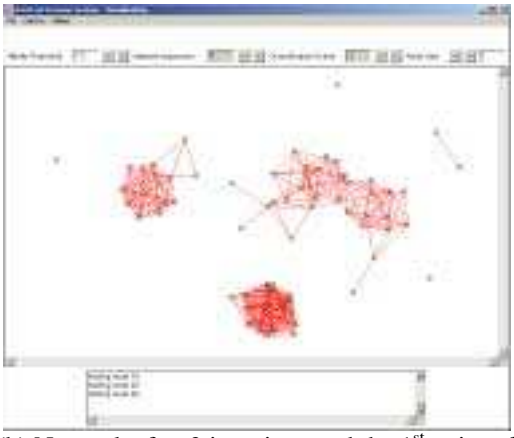


Figure 5.18: Core Population and average linkage for iterations 0 to 99 learning the iris data set. Mutation rate 0.05, NAT Scalar 0.25 and a Maximum Number of Resources 480.

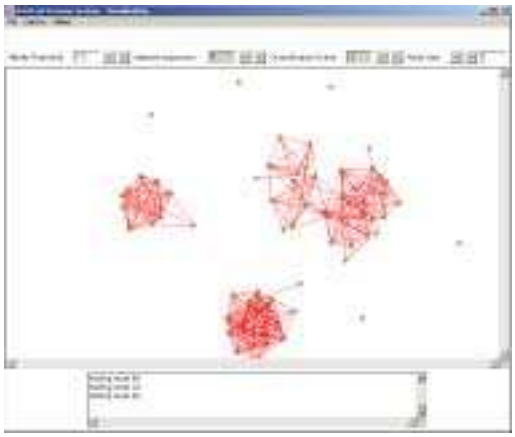
The goal of the AINE algorithm is to learn the data set being analysed and produce reduced complexity representations of the data space, and for these reasons only the stable patterns in the first 20 iterations are going to be considered. It is during this period that the average linkage remains low and stable. The screenshots in figure 5.19 show the evolution of the network at six points between iteration 1 and 20, including the two stable periods. It can be seen that after 11 iterations the Setosa class starts to degenerate until it is completely gone after iteration 20. The Virginica and Versicolor classes become joined and inseparable. The most likely cause for this is the production of clones that have mutated to a point that they represent parts of both classes.



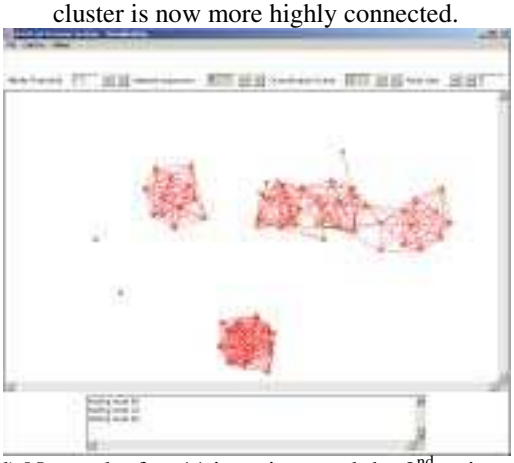
(a) Network after 1 iteration all three classes are present, each cluster has a fairly sparse linking structure.



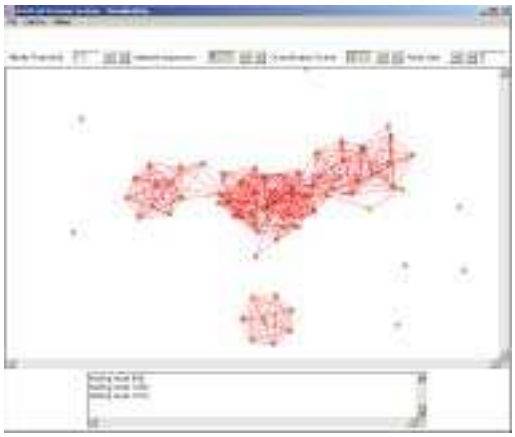
(b) Network after 3 iterations and the 1st point of stability (see Figure 18). The structure of each cluster is now more highly connected.



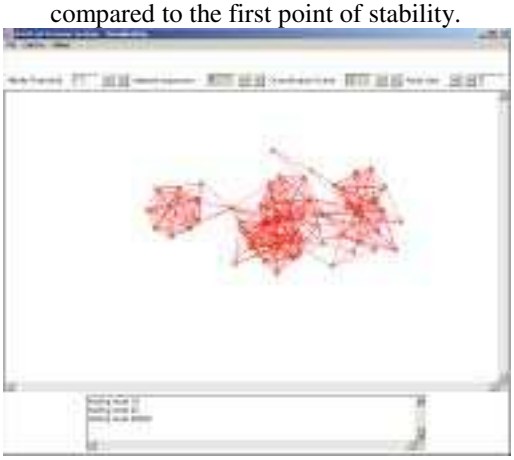
(c) Network after 7 iterations and the Versicolor cluster (top-right) appears to be growing.



(d) Network after 11 iterations and the 2nd point of stability. The Versicolor cluster has grown larger compared to the first point of stability.



(e) Network after 17 iterations. The Setosa cluster (bottom) is becoming more sparse and the Virginica and Versicolor clusters are now connected.



(f) Network after 20 iterations. The Setosa class has completely disappeared from the network, and the Virginica and Versicolor classes are still connected.

Figure 5.19 (a-f): Network evolution of the iris data set viewed using aiVis, Virginica; top-left, Versicolor; top-right, Setosa; bottom-middle.

5.3.5.1 Trends in the Iris Data set

The trends in the Iris data set were also analysed to provide a clear understanding of what happens to each class during a run of the algorithm. Figure 5.20, is the trend plot from the run described above. From this plot it is clear to see that after iteration 11 there is a steep decline of the Setosa class. This plot also shows that the Versicolor class is dominant throughout the learning process, with almost two times as many ARBs in the network after the Setosa class is removed.

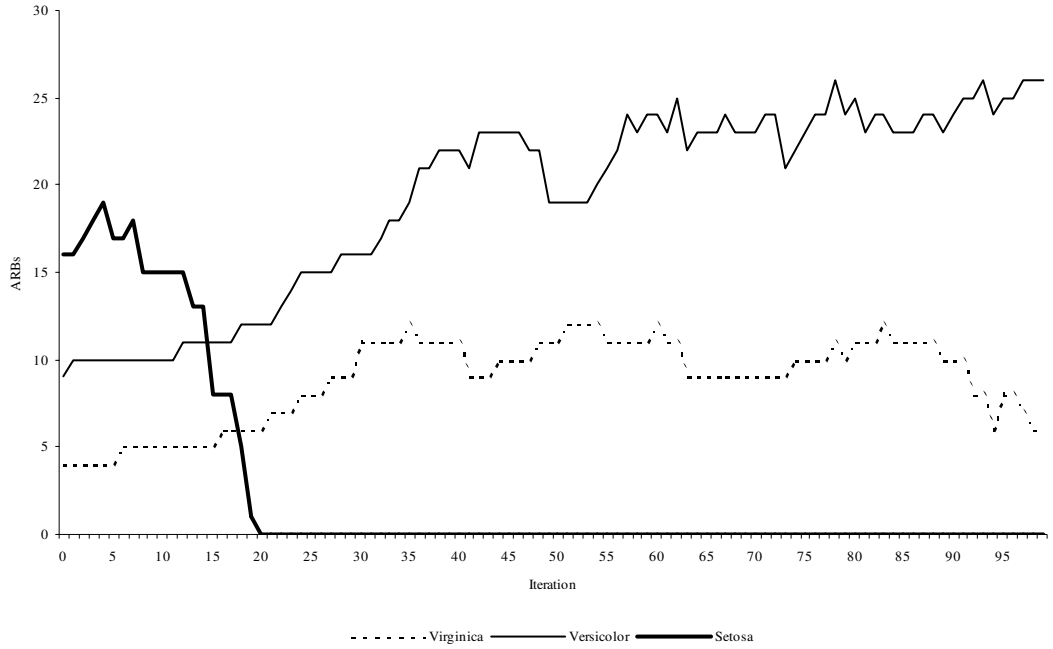


Figure 5.20. Network Trends for the iris data set with a Mutation rate of 0.05, NAT Scalar of 0.25 and a maximum number of resources of 480.

5.3.5.2 Summary of Network Evolution

AINE was run using the following settings; Mutation rate of 0.05, NAT Scalar of 0.25 and a Maximum number of Resources 480 for 100 iterations. Stable networks were produced at iterations 3-4, 10-11, 51-52 and 66-67 of which the first two were analysed. The first two periods of stability showed good separation of the three classes but following the 20th iteration the network structure began to degrade and only the strongest classes survived. This is again evidence that the algorithm is capable of discovering the core patterns, but it still showing problems in maintaining long-term representation of the core structures.

5.4 Summary

The revised version of AINE is tested using the same strategy as in (Timmis, 2001) to ensure continuity with results. The two data sets used are a simulated data set, and the machine learning benchmark, the Iris data set (Fisher, 1936). Tests were run on three different parameters, the Mutation Rate, the NAT Scalar and the maximum Number of Resources. The results were analysed based on two network characteristics, connectivity and size. The following is a brief summary of the results:

- The NAT Scalar affects the connectivity of the networks produced, for example a NAT Scalar close to the value 1.0 would produce very highly connected networks,

whereas a value close to 0.0 would produce very sparse networks. This property could be well used in reducing the complexity of large data set. The network size is not affected by this parameter.

- The Mutation Rate was shown to not affect either the network connectivity or network size but rather can be used to produce (using high values near 1.0) very diverse networks.
- The Number of Resources was shown to affect the size of the networks only, although it was found that for a given data set there was a minimum number of resources that could be applied in order for the resultant initial network to be representative of the data set to be learnt.

It was also observed that the revised algorithm tends towards the best solution / strongest class and therefore the possibility of continual learning in its present state with this version of the algorithm has been ruled out.

6 Conclusions

A full revision of AINE originally proposed by (Timmis, 2000) was undertaken and errors were identified and corrected. Following the corrections, the revised version of AINE exhibits slightly different behaviour to the original ANIE and therefore a comprehensive series of test were undertaken. Tests originally undertaken in (Timmis, 2001) were repeated with the following results:

- The NAT scalar can still be used to affect network connectivity and is an effective parameter in reducing complexity of data set. It has no effect on network size.
- The Number of Resources is an important control on network size, but does not affect the connectivity of the network. The greater the number of resources, the higher the chance weak ARBs have of surviving in the network.
- The Mutation Rate appears to have no affect on either the network connectivity or size. However it does appear to be a factor in determining the diversity of the networks and therefore the speed at which they deteriorate.

The results from these tests compare well with those in (Timmis, 2001), but following the correction in the algorithm the behaviour now appears to follow a more elitist strategy, where the networks will tend towards the strongest cluster, slowly removing the weaker clusters.

6.1 Future Work

As has been shown in this report, there are parts of AINE that can be improved. One particular direction that could be taken is the improvement of the mutation operation. The current operation is a very simplistic randomisation of one or more of the dimensions of the data item stored in the ARB. One possible avenue could be to try and use the somatic hypermutation operation as inspiration for an improved mutation algorithm. There is also the idea of using gene library from which it is possible to build initial populations. These are just a couple of ideas, but there are thousands that could be gleaned from the human immune system that may be of use.

7 References

de Castro, L.N. (1999) *Artificial Immune Systems: Part 1 – Basic Theory and Applications*. Technical Report, RT-DCA 01/99.

Coutinho, A. (1980). *The self-non self discrimination and the nature and acquisition of the antibody repertoire*. Annal Of Immunology (Inst. Past.) 131D.

Dasgupta, D., (1999), "*Immunity-Based Intrusion Detection System: A General Framework*", In Proc. of the 22 nd NISSC.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors (1996). *Advances in Knowledge Discovery and Data Mining*. MIT Press, Cambridge, MA.

R.A. Fisher (1936). *The use of multiple measurements in taxonomic problems*. Annals of Eugenics, 7-II: 179-188.

Forrest, S., Hofmeyr, S.A. and Somayaji, A. (1996). *Computer Immunology*. Communications of the ACM.

Hunt, J. E. and Cooke, D. E., (1996). *Learning using an artificial immune system*. Journal of Network and Computer Applications, 19:189—212.

Hunt, J. E. and Fellows, A. (1996), "*Introducing an Immune Response into a CBR system for Data Mining*", In BCS ESG'96 Conference and published as Research and Development in Expert Systems XIII.

Jerne, N. (1974). *Towards a network theory of the immune system*. Annals of Immunology (Inst. Pasteur). 125C, pp. 373-389

Kepler, T.B., Perelson, A.S. (1993). *Somatic hypermutation in B cells: an optimal control treatment.* J. Theor. Biol. 164, 37-64.

Nossal, G.J.V (1994) Life, Death and the Immune System: *Life, Death and the Immune System.* Scientific American, Special Issue. W.H.Freeman and Company. NY.

Perelson, A. (1989). *Immune Network Theory*. Immunological Review. 110, pp 5- 36.

Taranakov, A and Dasgupta. (2000) D. *A formal model of an artificial immune system.* BioSystems, Vol:55:151 –158.

Timmis, J. (2000a). *Artificial Immune Systems : A novel data analysis technique inspired by the immune network theory.* Ph.D. Thesis. University of Wales, Aberystwyth. 2000

Timmis, J. (2000b). *On parameter adjustment of the immune inspired machine learning algorithm AINE.* University of Kent at Canterbury. Technical Report 12-00.

Timmis, J. (2001). *aivis - artificial immune network visualisation*. In *EuroGraphics UK 2001 Conference Proceedings*, pages 61-69, Univerisity College London.

Timmis, J and Knight, T (2001). *Immunological Computation: Using the Immune System for Data Mining.* Accepted for publication in Data Mining a Heuristic Approach.

Timmis, J., Neal, M. and Hunt, J. (1999). *Data Analysis with Artificial Immune Systems, Cluster Analysis and Kohonen Networks: Some Comparisons.* Proc. Of Int. Conf. Systems and Man and Cybernetics, pages 922-927, Tokyo, Japan., IEEE

Timmis, J and Neal, M. (2000). *A resource limited artificial immune system for data analysis.* Research and Development in Intelligent Systems XVII. Pp 19-32.