Stable Clusters Formation in an Artificial Immune System

S.T. Wierzchoń

Department of Computer Science,
Białystok Technical University
ul. Wiejska 45[a], 15-351 Białystok, Poland
and
Institute of Computer Science,
Polish Academy of Sciences
ul. Ordona 21, 01-267 Warszawa, Poland

U. Kużelewska

Department of Computer Science,
Białystok Technical University
ul. Wiejska 45[a], 15-351 Białystok, Poland

Abstract

A new version of an artificial immune system designed for automated cluster formation in training data is presented. The algorithm fully exploits self-organizing properties of the vertebrate immune system and produces stable immune network. The algorithm uses the minimal number of control parameters.

1 INTRODUCTION

Artificial Immune System, or AIS, is a new, biologically inspired, paradigm of information processing. Its main principles are abstracted from the behaviour and properties of the vertebrate immune system, which is responsible for maintaining homeostasis of a living organism and particularly for protecting the organism from pathogens that could disrupt that homeostasis. More precisely, the immune system is a multi-layered structure. Each layer is of different complexity and the most complex is so-called *adaptive* immune system – consult (Hofmeyr, 2001) for details. In this paper by "immune system" we will understand the adaptive immune system. From a computer science perspective this last layer is a complex, self organizing and highly distributed system which has no centralized control and which uses learning and memory when solving particular tasks. The learning process does not require negative examples and the acquired knowledge is represented in explicit form.

The main actors of the adaptive immune system are B-lymphocytes (or B-cells) which mature in *bone marrow*, and T-lymphocytes (or T-cells) which mature in *thymus*. B-cells can be viewed as the commandos equipped with specialized weapon (i.e. antibodies attached to a single B-cell surface); each type of weapon is designed to fight different kind of enemy (i.e. pathogen or more precisely: antigen). B-cells can start their attack only after receiving signal from their commanders, i.e. subspecies of T-cells called helper T-cells, or Th-cells.

Thus, from a computer science point of view, Th-cells are responsible for Self/Non-self discrimination and the mechanisms governing Th-cells behaviour are used for designing novelty detection algorithms which can be used e.g. in the detection of computer viruses, or anomaly detection – consult (Dasgupta, 1999) for details. The "algorithms" used by B-cells (and reviewed in Section 2) are useful in adaptive data analysis, (Timmis, 2000), (De Castro and von Zuben, 2001), machine learning (Hunt and Cooke, 1996) or function optimisation (Bersini, 1990).

In this paper a new algorithm for adaptive clusters formation is given. Mentally based on the idea developed by Timmis (2000) the algorithm almost does not require control parameters and produces stable, long lived clusters. Section 3 describes this new algorithm and Section 4 contains numerical examples. Some general properties of the algorithm are discussed in Section 5.

A reader interested in models used in theoretical immunology is referred to the paper (Perelson and Weisbuch, 1997).

2 IMMUNE PRINCIPLES

Perhaps the first paper announcing exciting properties of the immune system was that of Farmer, Packard and Perelson (1986). As noted by Timmis (2000), the model described in this paper has shown how to use immune mechanisms in designing computer learning systems by: (i) using the idea of idiotypic network to achieve memory of what is being learnt, (ii) using a simple pattern matching mechanism between B-cell and antigen to define affinity, (iii) only representing B cells in the model and ignoring the effect of T cells, (iv) using a simple equation to model the stimulation of the B-cell, and (v) using mutation mechanisms to create diverse set of B cells. Let us briefly describe main mechanisms engaged during the immune response.

A single B-cell has about 10^5 receptors (antibodies) located on its surface. Each receptor has a specialized region, called *paratope*, used for identifying other molecules. Being a 3-D structure with uneven surface the paratope have a unique shape and other unique characteristics (e.g. van der Waals forces) referred to as the *specificity*. The regions on any molecule that the paratopes can attach to are called *epitopes*. If the two colliding molecules have complementary specificities,

they bind to each other and the strength of the bond (called *affinity*) depends on the degree of complementarity. A molecule bound by an antibody is referred to as the *antigen*[1]. A crucial role of the immune system is the binding of antibodies with antigens which serves to tag them for destruction by other cells. This process is termed *antigen recognition*. To treat formally the recognition problem, Perelson (1989) introduced the notion of the *shape space*. Namely, if there are m features influencing the interaction between the molecules (i.e. the spatial dimensions, charge distribution, etc.) and D_i is the domain of i-th feature ($i = 1, \ldots, m$) then each molecule is reduced to a point (the generalized shape of a molecule) in m-dimensional space ($S = D_1 \times \ldots \times D_m$. Typically S is a subset of m-dimensional Hamming space, or m-dimensional Euclidean space.

When a B-cell recognizes an antigen, it clones (i.e. produces identical copies of itself) as well as secretes free antibodies. The process of amplifying only those cells that produce a useful antibody type is called *clonal selection*, and the number of clones produced by a lymphocyte is proportional to its stimulation level. Clones are subjected to *somatic mutation* (characterized by high mutation rate) that results with new species of B cells having slightly different antibodies. These new B cells also bind to antigens and if they have a high affinity to the antigens they in turn will be activated and cloned. The rate of cloning a B-cell is proportional to its "fitness" to the problem: fittest cells replicate the most. The somatic mutation guarantees sufficient variation of the set of clones, while selection is provided by competition for pathogens. The whole process of (in fact: Darwinian) selection and differentiation of B-cell receptors leading to the evolution of B-cell populations better adapted to recognize specific epitopes is said to be *affinity maturation*.

Besides somatic mutation the immune system uses a number of other mechanism to maintain sufficient diversity and plasticity. Particularly about five percent of the B-cells are replaced every day by new lymphocytes generated in the bone marrow. This process is termed *apoptosis*.

The immune system possesses two types of response: primary and secondary. The *primary response* occurs when the B cells meet the antigen for the first time and reacts against it. To learn the structure of the antigen epitopes, clonal selection and somatic mutation are used. The primary response takes some time (usually about 3 weeks) to destroy the antigen. If the organism is reinfected with a previously encountered antigen, it will have an adapted subpopulation of B-cells to provide a very specific and rapid *secondary response*. From a computer science perspective the primary response corresponds to the identification of clusters in the training data, while the secondary response – to the pattern recognition problem, i.e. the assignment of a new data into one of existing clusters. Interestingly, the secondary response is not only triggered by the re-introduction of the same antigens, but also by infection with new antigens that are similar to previously seen antigens. That is why we say that the immune memory is *associative*. This phenomenon is modelled in the shape-space formalism by introducing so-called *recognition ball*, i.e. a ball B_r with radius r and centred in the point corresponding to the generalized shape of a given antibody.

The final immune system principle that plays a useful role in designing AIS's is that of *immune network* theory formulated by Jerne (1974), and further developed by Perelson (1986). According to this theory (called also *Jerne's hypothesis*) the immune response is based not only on the interaction of B-cells and antigens but also on the interactions of B-cells with other B-cells. These cells provide both a stimulation and suppression effect on one another and it is partially through this interaction that the memory is retained in the immune system.

The immune system is in permanent flux. The whole network is subjected structural perturbations through appearance and disappearance of some cell species. The introduction of new species is caused by somatic mutation, apoptosis, or combinatorial diversity (e.g. genetic operations used to produce new paratopes). A crucial issue is the fact that the network as such, and not the environment, exerts the greatest pressure in the selection of the new species to be integrated in the network. Thus, the immune network is self-organizing, since it determines the survival of newly created clones, and it determines its own size. This is referred to as the *meta-dynamics* of the system, (Varela and A. Coutinho, 1991).

The two most influential data analysis systems based on the immune metaphor are `aiNet` (De Castro and von Zuben, 2001) and `AINE` (Timmis, 2000). In both the systems the training set is identified with the set of antigens and the aim is to produce a set of B cells or antibodies representing these antigens.

According to Jerne's hypothesis, `AINE` produces networks (counterparts of idiotypic network) describing the key features of data items within the training set. The system uses almost all mechanisms described in this section, i.e. (i) it uses a set of B cells each of which is capable of recognizing antigens, (ii) similar B cells are linked together; these links form a network of B cells, (iii) clonal selection and hypermutation are performed on B cells, (iv) a number of B cells can be represented by an artificial recognition ball, or ARB. In fact, to improve stability of the immune network, AINE uses a population of ARBs and not the population of B cells. It needs four important control parameters: network affinity threshold (NAT), the mutation rate, the number of ARBs and the number of clones produced by a stimulated ARB. The influence of these parameters on final network is analysed in (Knight and Timmis, 2001).

The `aiNet` system, on the other hand, uses simplified representation: instead of B cells or ARBs it simply

[1] Antigen is a shorthand of *anti*bodies *gen*eration.

develops a population of antibodies. The population is initialised randomly (while AINE uses a random subset of antigens) and next it is modified by clonal selection, hypermutation and apoptosis. Interesting feature of the algorithm is that the clonal selection controls the network dynamics and metadynamics. Its main drawback is large number of user-defined control parameters. Further to obtain the immune network we have to use standard clustering tools: hierarchical clustering and graph-theoretic algorithms. But its advantage is very concise description of training data. In some cases such a data reduction equals 90% (Wierzchoń, 2001).

Both the algorithms are examples of *unsupervised* machine learning algorithms. Watkins (2001) used a combination of the just described approaches to design *supervised* learning algorithm. His aim was to develop a predictive model based on input data and the known classes in the data set.

To finish this section, let us note that the model of immune memory proposed by Jerne resembles the models of hypercycles or autocatalytic sets considered in the context of prebiotic chemical evolution – cf. (Bagley and Farmer, 1992) or (Eigen, 1971). It seems that careful examination of these models may be of value in constructing effective data analysis.

3 A NEW ALGORITHM

As stated in previous section, natural immune system contains B-cells with antibodies attached to their surfaces.

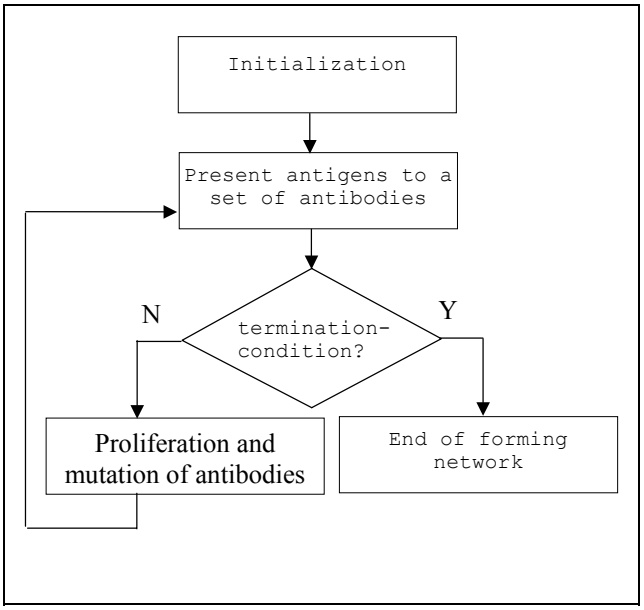

Figure 1. The algorithm for an immune network generation

In our AIN (artificial immune network) B cells are reduced to antibodies. This resembles the idea applied in the aiNet system. Let $\mathbf{Ab} = \{\mathbf{ab}_1, \ldots, \mathbf{ab}_n\}$ denotes the set of antibodies, and $\mathbf{Ag} = \{\mathbf{ag}_1, \ldots, \mathbf{ag}_k\}$ be the set of antigens. Each element $\mathbf{y} \in \{\mathbf{Ab}, \mathbf{Ag}\}$ is m-dimensional real-valued vector $\mathbf{y} = \{y_1, \ldots, y_m\}$.

The algorithm depicted on Figure 1 creates an AIN in the way similar to that used in (Timmis, 2000) but with some significant modifications. The nodes of the AIN represent antibodies, and their aim is representation of generalized characteristics of the antigens. Connected antibodies form clusters, so if any antibody from the cluster recognizes an antigen, it means the antigen belongs to this cluster. Recognition of an antigen, **ag**, by antibodies relies upon searching for an antibody **ab*** that minimizes Euclidean distance $d(\mathbf{ag}, \mathbf{ab}_i)$, $i = 1, \ldots, n$. The inverse of $d(\mathbf{ag}, \mathbf{ab}_i)$, can be viewed as the *affinity* of **ag** to $\mathbf{ab}_i$. Thus the smaller the distance $d(\mathbf{ag}, \mathbf{ab}^*)$ the better representation of **ag** by **ab*** is. The maximal length of an arc joining two nodes in the AIN is just the *NAT* scalar. Cells (antigens and antibodies) located further than *NAT* do not influence one another. The *NAT* parameter determines the granularity of the network and its overall connectivity (Knight and Timmis, 2001).

3.1 INITIALIZATION OF THE AIN

Like in (Timmis, 2000) this process is divided into three stages. First, the training data are normalized, i.e. the set **Ag** becomes a subset of m-dimensional unit cube $[0, 1]^m$. The second stage is to calculate an initial NAT value using the **Ag** set. Timmis calculates it as $\langle d \rangle \cdot \alpha$, where $\langle d \rangle$ is the average distance between each item in **Ag** and $\alpha \in (0,1\rangle$ is a constant. Such computed parameter was fixed during whole process of network creation. However such a procedure requires several runs of the algorithm and necessity of choosing the best value. In our approach the NAT value is computed in every iteration of the algorithm without external intervention. Initial NAT value is computed as follows. Let

$$D = \{d(\mathbf{ag}_i, \mathbf{ag}_j): i = 1,\ldots, k\text{-}1,\ j = 2, \ldots, k, j > i\}$$

be the set of distances values between each unique pair of antigens. Sort ascending the elements of D and denote

$$D' = \{d^1, \ldots d^l\} \quad (1)$$

a subset of D consisting of $l \le k \cdot (k-1)/2$ initial elements. Now the NAT is computed as the average value of the distances in the set D'. The third stage is to construct initial immune network. That is the set of antibodies **Ab** (also a subset of $[0, 1]^m$) is *randomly* initialised. This is in contrast with Timmis' approach. Next two antibodies are joined together only if their distance is not greater than the NAT value.

3.2 PRESENTATION OF THE ANTIGENS

In this stage every member of the **Ag** set is presented to the network and the stimulation level of each antibody, $sl(\mathbf{ab}_i)$, $i = 1,\ldots, n$, is computed.

According to Jerne's hypothesis the stimulation level of a B-cell is the sum of three factors: its affinity to the antigens, its affinity to its neighbours in the network and its enmity to these neighbours. This definition of the

stimulation level was applied in the AINE system. On the other hand, in the aiNet only the affinity of each antigen to all antibodies was taken into account. Similar idea was implemented in our system AIN. Define namely $\delta_i = \min_j d(\mathbf{ab}_i, \mathbf{ag}_j)$ to be the minimal distance between i-th antibody and the set of antigens. Now if $\delta_i \le$ NAT then $sl(\mathbf{ab}_i) = 1 - \delta_i$, and $sl(\mathbf{ab}_i) = 0$ otherwise.

Knowing the stimulation level of each antibody we can implement apoptosis, i.e. we can define the set of *effective* antibodies, $\mathbf{Ab}^* \subseteq \mathbf{Ab}$. Initially $\mathbf{Ab}^* = \mathbf{nil}$. First of all antibodies with zero stimulation value are removed from the set **Ab**. Denote **Ab**′ reduced set. Next, for each antigen, **ag**, we determine the set of antibodies recognizing this **ag**. If a given antigen is recognized by the unique antibody **ab*** then add this **ab*** to the set **Ab***. Hence **Ab**′ – **Ab*** is the set of potentially redundant antibodies. If $\mathbf{Ab}'' \subset (\mathbf{Ab}' - \mathbf{Ab}^*)$ is the set of antibodies each of which recognizes a group of identical antigens, we find a single antibody with highest stimulation value; only this antibody is moved to the set **Ab***.

This way we are still in the frames of an idiotypic network. Stimulation value awards antibodies with highest affinity to the antigens, while suppressive mechanisms are moved to the purging procedure. It seems that correctly designed purging procedure is responsible for generation of stable immune networks. Nasaroui, Gonzales and Dasgupta (2002) introduced fuzzy ARBs to improve stability of the immune networks. In our opinion it is not necessary. We can even use "standard" definition of stimulation level as proposed by Jerne and we can still generate stable networks provided that efficient antibodies elimination (described above) is implemented – see Sect. 4 for numerical results.

3.3 PROLIFERATION

Again this process is divided into three stages. First, the NAT value is recalculated using the cells from the set **Ab***. To do so, the set D' – see Eqn. (1) – is constructed; its cardinality is $l' \le l$.

Second, most stimulated cells from the set **Ab*** are cloned and mutated. In cloning process an antibody with stimulation level sl produces $\lfloor c_{max} \cdot sl \rfloor$ clones, where c_{max} is a constant (maximal number of clones). Clones are added to a separate set of clones **C**. Each clone $\mathbf{c} = (c_1, \ldots, c_m)$ is subjected mutation according to the equation

$$c_i = c_i + r \cdot \Delta,\ i = 1, \ldots, m$$

where r is a random number from the unit interval and $\Delta = 1 - y_i$ or $\Delta = -y_i$ (the decision which Δ value to choose is made randomly).

Third, mutated clones from the set **C** are integrated with the network, i.e. $\mathbf{Ab} = \mathbf{Ab}^* \cup \mathbf{C}$. Finally the immune network is reconstructed: two antbodies $\mathbf{ab}_i$, $\mathbf{ab}_j$ are joined together only if $d(\mathbf{ab}_i, \mathbf{ab}_j) \le$ NAT.

4 EXPERIMENTS

To verify the quality of this new algorithm, three data sets were analysed. In each experiment we focused on two-dimensional data representing two separate clusters. The first experiment is concerned with linearly separable clusters (Figures 2a-2d). In two remaining experiments, Figures 3a-3d and 4a-4d, training data exhibiting non-trivial patterns were used. Every AIN was developed through 50 iterations to observe stabilization of the NAT value as well as stabilization of resulting network size.

Every set of figures, denoted a – d, includes: (a) antigen set, (b) final network structure, (c) evolution of NAT and (d) evolution of the network size. In every case after some number of iterations network becomes stable. This number depends on size and complexity of antigen set.

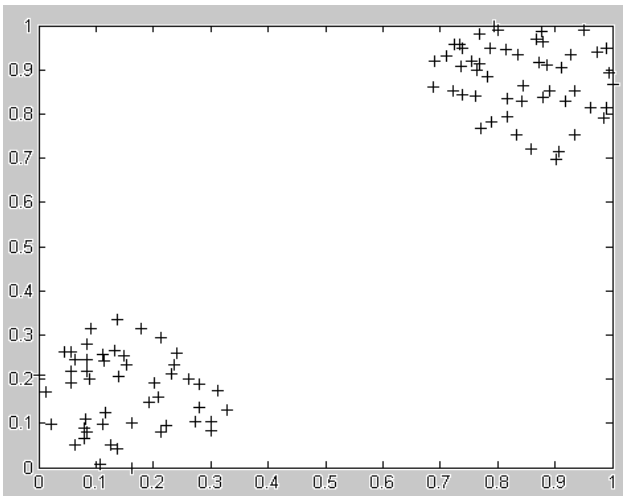

Figure 2a: Antigens set

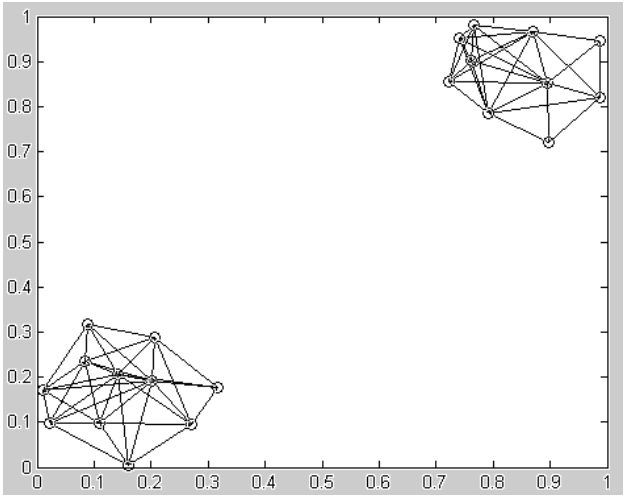

Figure 2b: Final immune network

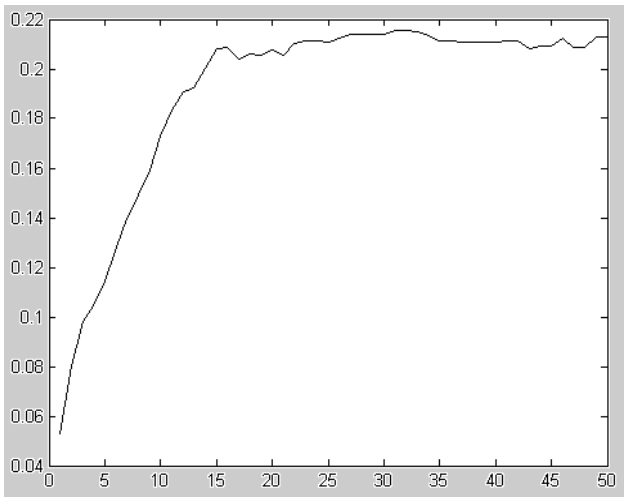

Figure 2c: Evolution of the NAT value

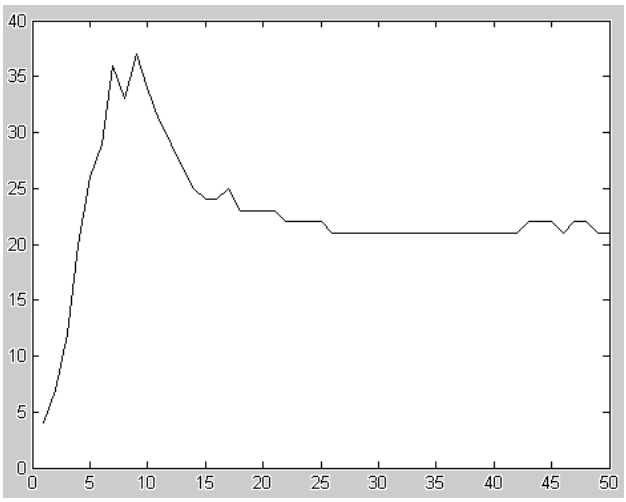

Figure 2d: Evolution of the network size

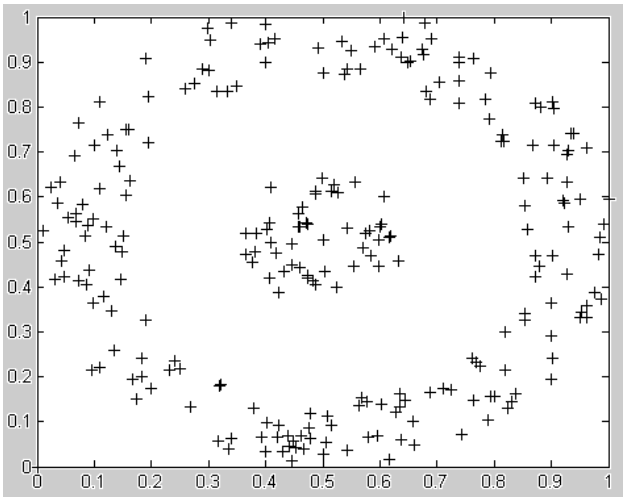

Figure 3a: Antigen set

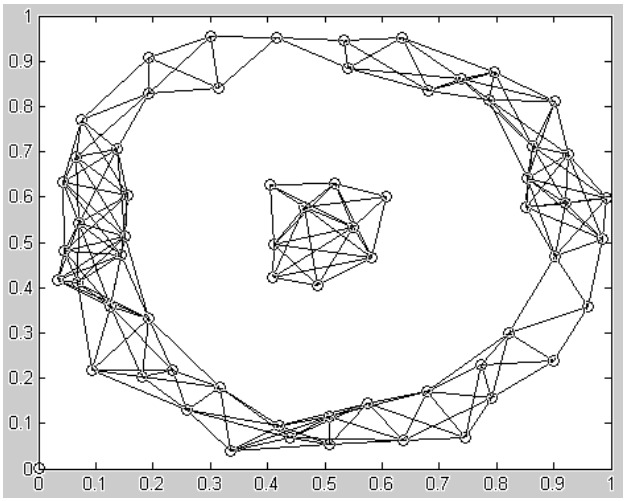

Figure 3b: Final immune network

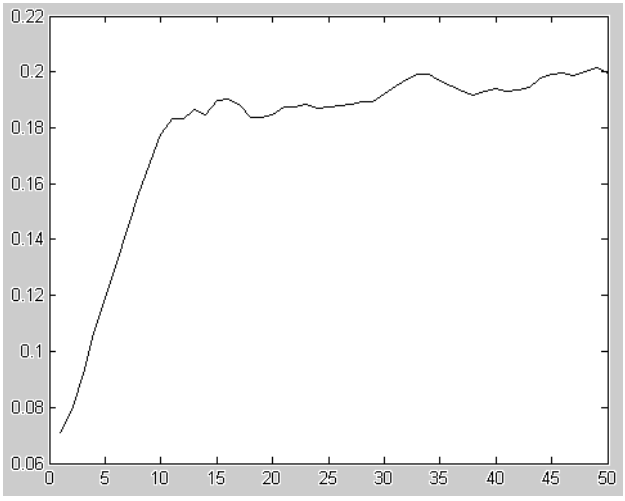

Figure 3c: Evolution of the NAT value

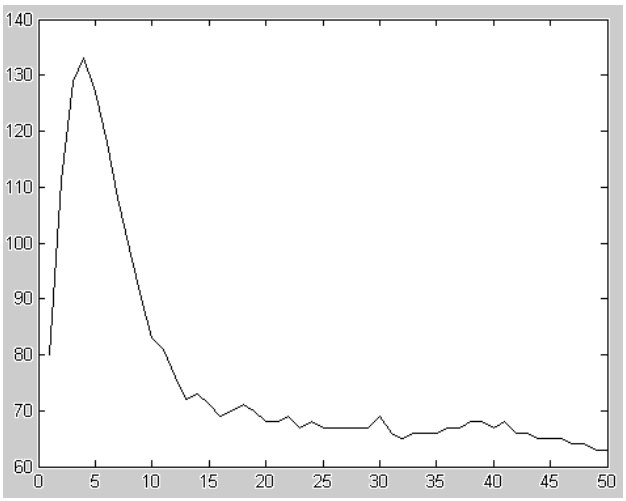

Figure 3d: Evolution of the network size

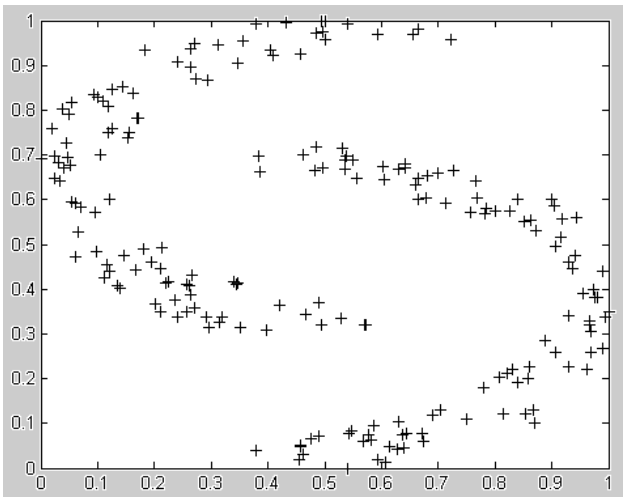

Figure 4a: Antigen set

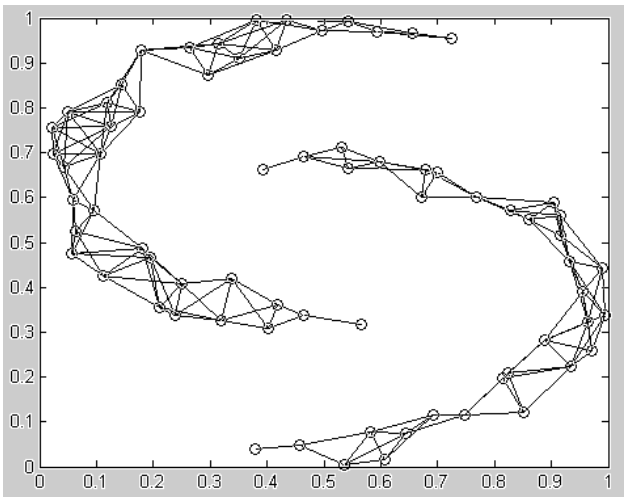

Figure 4b: Final immune network

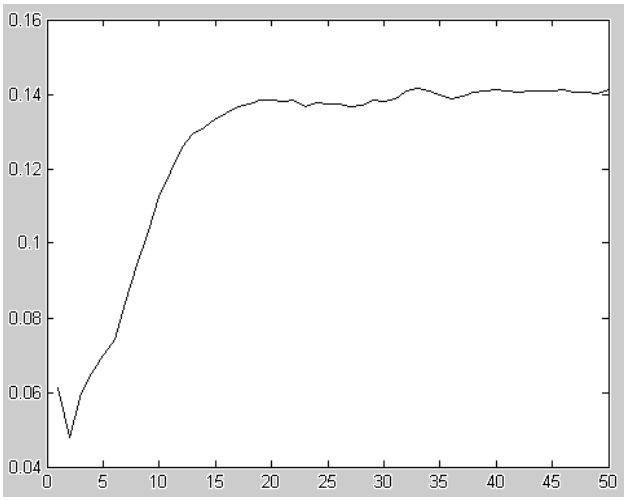

Figure 4c: Evolution of the NAT value

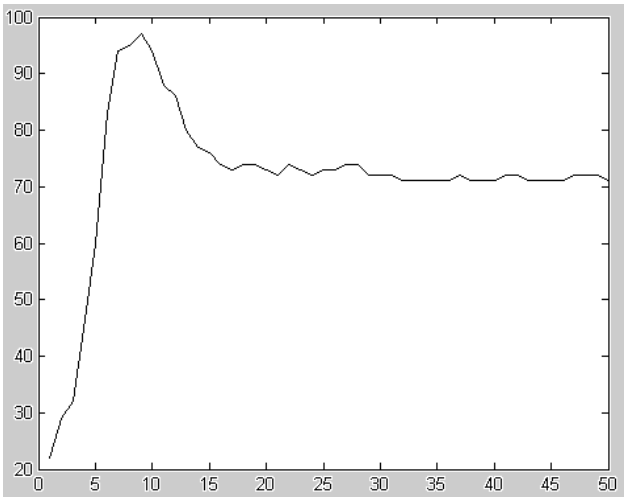

Figure 4d: Evolution of the network size

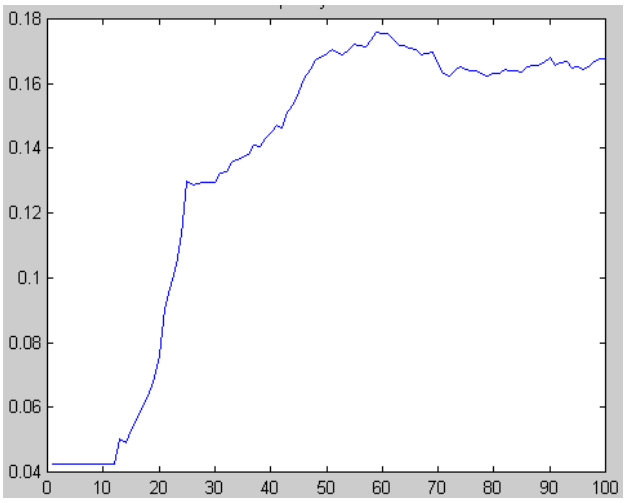

Figure 5a: Evolution of the NAT value in a system with "standard" definition of stimulation level

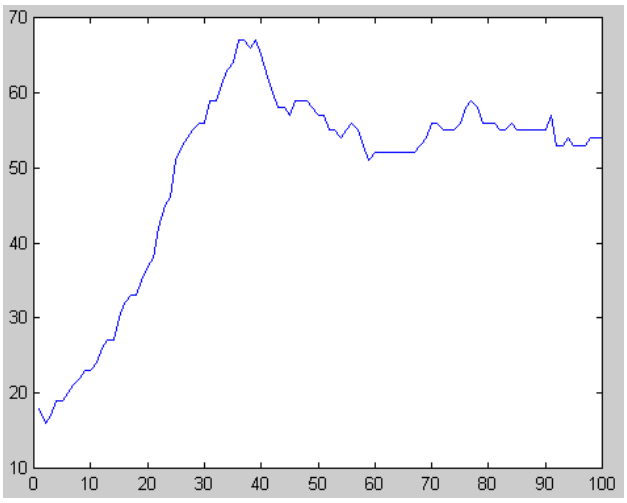

Figure 5b: Evolution of the network size in a system with "standard" definition of stimulation level

Interestingly, the algorithm behaves almost identical when the stimulation level is defined as in (Timmis, 2000). Figures 5a and 5b demonstrate evolution of the NAT value and network size for the antigen set presented

on figure 4a. In fact minor modifications of the purging procedure results in different behaviour of the algorithm (see the url: http://www.ipipan.waw.pl/~stw/ais for different data sets analysed with different purging strategy).

5 CONCLUSIONS

In all the cases analysed the algorithm is able to produce correct networks after 15-20 iterations. After this time the network structure becomes stable – its size and the NAT value oscillates around fixed value. This fact can be used as the definition of the termination condition. Final network structure represents *immune memory* – it can react faster and better when similar data are encountered in the future.

Interestingly, in each case we can observe data compression phenomenon like in the `aiNet` system. In first experiment the antigen set consists of 100 items and final immune network consists of 21 cells; so data compression ratio is 79%. In the second case data compression attains 69%, and in the third case – 29%. This property results from the second stage of the purging procedure (i.e. further reduction of the set **Ab′** described in Section 3.2). The compression ratio depends on the topology of input data.

The most important feature of the immune system is its ability to recognize new pathogens. The algorithm described in this paper passes this examination very well. The number, shape and location of generated clusters precisely reflects topological properties of the training set.

Additionally clusters are formed adaptively. This is in contrast to the aiNet system, where antibodies are generated first, and next graph theoretical methods are used to cluster these antibodies.

Finally the algorithm requires minimal number of control parameters indeed: it is necessary to define only the cardinality of the set D' – cf. Eqn. (1) – and maximal number of clones c_{max}. The NAT value evolves during subsequent iterations of the algorithm. The same applies to the network size.

6 FUTURE WORK

The algorithm described in this paper possesses many intriguing properties. Detailed mathematical analysis is necessary to confirm these properties. Particularly deeper analysis of the influence of stimulation level computation and purging implementation on the algorithm behaviour should be performed.

It was also observed (results not reported here) that in case of overlapping clusters the algorithm displays a kind of cross-reactive memory; this phenomenon should also be carefully verified.

Lastly, the algorithm behaviour on more complex data sets will be examined, and more flexible strategies for choosing effective antibodies **Ab*** will be worked out.

Acknowledgments

First author would like to thank Leandro Nunes de Castro and Jonathan Timmis for exhaustive comments on their systems as well as to the unknown referees for constructive remarks.

References

R. J. Bagley, and J. D. Farmer (1992). Spontaneous emergence of a metabolism. In: C.G. Langton, C. Taylor, J.D. Farmer, S. Rasmussen, eds., *Proc. of the Workshop on Artificial Life* (*ALIFE'90*), vol. 5 of *Santa Fe Institute Studies in the Sciences of Complexity*, Addison-Weseley 1992, 93-140

H. Bersini (1990) Hints for adaptive problem solving gleaned from immune networks. In: *Proc. of the First Workshop on Parallel Problem Solving from Nature*, LNCS 496, 343-354. Berlin: Springer-Verlag.

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

L. N. De Castro, and F. J. von Zuben (2001) aiNet: An artificial immune network for data analysis. In: H.A. Abbas, R.A. Sarker, Ch.S. Newton, eds., *Data Mining: A Heuristic Approach*, 231-259. Idea Group Publishing, USA

M. Eigen (1971) Selforganization of matter and the evolution of biological macromolecules. *Naturwis-senschaften* **58** : 465-523

J. D. Farmer, N. H. Packard, and A. S. Perelson (1986) The immune system, adaptation, and machine learning. *Physica D* **22**:187-204

S. A. Hofmeyr (2001) An interpretative introduction to the immune system. In L.A. Sagel and I.R. Cohen (eds.) *Design Principles for the Immune System and Other Distributed Autonomous Systems*, 3-26. New York: Oxford University Press.

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

N. K. Jerne (1974) Towards a network theory of the immune system. *Ann. Immunol* (*Inst. Pasteur*) **125C**: 373-389

T. Knight, and J. Timmis (2001) Assessing the performance of the resource limited artificial immune system AINE. Technical Report No. 3-01. Computing Laboratory, University of Kent, Canterbury, Kent, UK.

O. Nasaroui, F. Gonzales and D. Dasgupta (2002) The fuzzy artificial immune system: Motivations, basic concepts, and application to clustering and Web profiling. In *Proc. of the IEEE International Conf. On Fuzzy Systems at WCCI*, pp. 711-716, Hawaii, May 12-17

A. S. Perelson (1989) Immune network theory. *Immunological Reviews*, **110**: 5-33

A. S. Perelson, and G. Weisbuch (1997) Immunology for physicists. *Reviews of Modern Physics*, **69**: 1219-1265

J. I. Timmis (2000) Artificial immune systems: A novel data analysis technique inspired by the immune network theory. Ph. D. Thesis. Department of Computer Science, University of Wales, Aberystwyth, September 2000

F.J. Varela, and A. Coutinho (1991) Second generation immune networks. *Immunology Today*, **12**: 159-166

A. B. Watkins (2001) AIRS: A resource limited artificial immune classifier. M. Sc. Thesis. Department of Computer Science, Mississippi State University, December 2001

S. T. Wierzchoń (2001) *Artificial Immune Systems. Theory and Applications* (in Polish). Akademicka Oficyna Wydawnicza EXIT, Warszawa 2001. ISBN 83-87674-30-3, 282+vii pp.