The application of artificial immune system for parallel process of calculation and their comparison with existing methods

I F Astachova, S A Ushakov, A I Shashkin and N V Belyaeva
Department of Applied Mathematics and Mechanics, Voronezh State University, Voronezh, Universitetskaja Ploschad, 1, Russia
E-mail: astachova@list.ru

Abstract. The present paper is devoted to the application of artificial immune system method for the solution of the symbolic regression problem. A mathematical model with the algorithm of its realization in sequential and parallel realization using some properties of the natural immune system is proposed. In the description of the proposed artificial immune system model and algorithm the terminology of the natural immune system is preserved. The concept of B-cells, which are responsible for producing antibodies, a special class of complex proteins on the surface of B-lymphocytes, capable of binding to certain types of molecules (antigens), is adopted and used. Accuracy of calculations is compared to that for other methods, i.e. methods of genetic programming. The program product involving the algorithms is developed. Computing experiments for each considered task are made.

1. Introduction
Biological studies have become an inspiration and source for the development of many new algorithms, systems, and models in informatics, i.e., networks and evolutionary algorithms in particular. A neural network approach has influenced the methods of nonlinear modeling and pattern recognition, bringing the idea of a multi-level hierarchical decomposition and parallel data processing based on biological representations of brain functioning model. Evolutionary algorithms have particularly contributed towards the following: methods of parallel search, in which mechanisms of selection and generation of new candidates at random are taken place, and the use of recombination ideas (which help to construct a new candidate on the basis of already existing ones) in solving the optimization problems [1–3].

In [4] such basic quality of the biological immune system as double plasticity has been named to be the principle concept for the creation of algorithms for solving practical problems of various kinds. Lets look at double plasticity, what is it and what practical benefits it gives.

There are biological systems, which are capable of adaptation based on the parametric and structural changes, i.e., on double plasticity – parametric and structural. Parametric plasticity is a mechanism of adaptation, which allows the system during its operation to change its settings thus to improve the performance. An example of such plasticity would be a change of the weighting coupling coefficients in the neural network, which is an integral part of the nervous system functioning. Structural plasticity in its turn, offers the system new opportunities for adaptation, for example, if the system consists of interacting elements, then its structural
plasticity is the ability to remove existing elements and add new ones to it. For the neural networks it is the ability of neurons to appear or disappear at the right time, thus to change not only the weighting coefficients, but also the number of neurons in the network structure during the network operation.

The immune system also has a double plasticity and consists of interacting elements. The concentration of already existing elements in it changes through the mechanisms of birth and death of cells that compose the immune system. Another example of the system with double plasticity is ecosystem. Its parametric plasticity lies in variation of the number of different species, which happens as a result of their interaction with each other and with the environmental factors. New species in ecosystem appear due to the recombination and mutation of genes. However, it is important to note the major difference between the immune system and ecosystem. Immune system has the objective function and the global task – to protect the entire body. In ecosystem, on the contrary, each element is trying to improve their own behavior, thus to improve the individual status. The remaining elements of ecosystem can help or hinder the survival of the species, but the goal is not the survival of entire system as a whole. The function of the immune system is a collective and strives to preserve the whole system (one organism).

The problem, in which the concept of double plasticity could be used to help finding a more effective solution, is the problem of symbolic regression. In this problem, numerical coefficients in the searched function correspond to parametric plasticity, and the type and superposition of the searched function correspond to structural plasticity. Lets consider the problem of symbolic regression, and how it can be solved using artificial immune systems. There is a set of values of free variables \{x_1, \ldots, x_r\}, where \(x_i \in \mathbb{R}^n\) and corresponding to them values of function \{y_1, \ldots, y_r\}. These two sets create a set of source data – \(D\).

Also there is a given set of functions, which will be used for the construction of superposition. Let us consider here only continuously differentiable functions: \(g : \mathbb{R}^n \rightarrow \mathbb{R}\) such as \(\sin, \cos, \text{Polynomials}\).

We consider any superposition of function \(f\) consisting of \(m\) of functions \(g\). It is required to find such superposition, which would provide maximum or minimum for the functional \(p(f, D)\). This functional defines objective function. For the construction of superposition it is possible to use various metrics and distances, the relative and absolute error – the difference of initial values and the values of the resulting function at the same points.

Thus, the input data for the calculation is the set of points of \(n\)-dimensional space and values of function in this point. The solution of this task is the function depending on \(n\) variables, which approximates initial function the best way [5–9].

2. The structure of artificial immune system

We suggest that the mathematical statement of artificial immune system could be presented as a set of the following elements [10]:

\[ IIS = \langle L, G, A, m, S \rangle, \]

where:
- \(IIS\) is artificial immune system;
- \(L\) is the space of all possible lymphocytes, the lymphocyte can represent a line, the list of coordinates, an expression tree;
- \(G\) is a set of all possible anti-genes, in so doing \(G\) can be the line, a matrix of logical values, and/or a list the list of values of a function in the known points;
- \(A : L \times G \rightarrow [0, 1]\) is the given measure of affinity which assigns to each lymphocyte and each anti-gene a certain number from the segment \([0, 1]\) (this number shows how "well" this lymphocyte reacts on the given anti-gen;
\( \mu : L \rightarrow L \) is the mutation operator, which is applied to a separate lymphocyte for the improvement of its recognition property;

\( S : A \subset L \rightarrow B \subset A \subset L \) is the selection operator leaving the best lymphocytes in the current immune system, supporting the network size.

Then algorithm could be presented as a sequence of the following steps:

- Step 1 is to create initial immune system –
  \[ ImSystem \subset L. \]

On this step, the given number of admissible lymphocytes for a particular problem is generated randomly to form an initial system.

- Step 2 is to get
  \[ g \in G, \ \forall l \in ImSystem : a_l = A(l, g). \]

On this step, the corresponding antigens approach to all lymphocytes of the current immune system, and affinity (fitness) is calculated.

- Step 3 is to define the best lymphocyte
  \[ l^* = \arg \max(a_l). \]

- Step 4 is to apply the mutation operator \( M = \{ \mu(l), l \in ImSystem \} \) to lymphocytes.
  The mutation operator may be applied not to all lymphocytes, but to a certain subset (more frequently to those possessing the higher value of affinity). The mutation operator inserts little changes in the value or structure of a lymphocyte.

- Step 5 is to utilize the selection operator
  \[ ImSystem = S(ImSystem \cup M), \]

which selects and remains lymphocytes with the greatest values of affinity from the current set of lymphocytes and from set of the mutated lymphocytes obtained on step 4.

- Step 6 is to decide whether \( l^* \) satisfies to the given criterion or whether the maximum number iteration is achieved. If yes, then go to the exit, otherwise return to Step 2.

Summarizing, it could be noted that the immune system solves a problem of function optimization that represents affinity. For various technical tasks this model and algorithm could be changed depending on the problem under consideration.

1. The first task is to define how the lymphocytes will be presented. Since we have a function as a solution of the problem, i.e. superposition of the defined simple functions, then the lymphocyte should be this function, encoded in a certain way. A convenient presentation of the lymphocytes for direct implementation, as well as for the further work of the algorithm, could be the expression tree. This tree will be binary by means of binary and unary operations. For example for the expression \( x-2*(1/x+x/3) \) the tree will look like the following (figure 1) [11–14].

Since the number of variables is known beforehand, a lymphocyte also stores a list of valid variables.

All lymphocytes need to support the following operations:

- Calculation of the value of the given function at a point.
- Calculation of the affinity of the given lymphocyte (discussed below).
- Simplification of the corresponding expression.
- Return of the line representation of expression (with the correctly placed brackets)
Thus the function presented by the lymphocyte can be written as [5–9]:

\[ F = f_1(f_2(\ldots f_m(x_1, x_2, \ldots, x_n))), \]

where \( f_1, f_2, \ldots, f_m \) – functions from the given set, \( m \) – number of functions, less than or equal to the maximum allowable height of an expression tree, \( x_i \) – free variables.

2. Affinity (objective function). Let us introduce the concept of affinity. In biology, affinity of antibodies is the binding strength of the active centers of the antibody molecule with antigen determinant groups. The antigen is a set of input data – values of functions at given points. In the given model affinity characterizes the degree of "success" of approximation, i.e. represents the value of the objective function. Let us use the following function:

\[ \sqrt{\sum_{i=1}^{r} f(x_i - y_i)^2}, \]

where: \( \{x_1, \ldots, x_r\}, x_i \in R^n \) – a specified set of values of free variables; \( \{y_1, \ldots, y_r\} \) – a specified set of values of the search function at appropriate locations \( x_i \);

\( f(x_i), i = 1, r \) – multiple values of the function represented by the given lymphocyte in corresponding locations \( x_i \).

- The value of the function at a point would be calculated;
- The measure of affinity would be calculated;
- The corresponding expression would be simplified;
- The representation of expression would be returned with correctly placed brackets.

In immunology, affinity is a measure of how well selected B cell interacts with the antigen. In our model affinity is a measure of how close the function (represented by the lymphocyte) is to solution. We present the affinity function using the Euclidian metrics.

3. Immune system or immune network consists of many lymphocytes. Because of decentralized nature of the immune system, artificial immune systems can have many groups of lymphocytes, which can be placed on different computational nodes. These groups of lymphocytes can communicate with the others, share good solutions and maintain a variety of solutions.
3. The algorithm of artificial immune system

Number of algorithms can be found in [15]. We use the following algorithm:

**Step 0.** The system of initial set of lymphocytes is generated randomly. Some predefined number of lymphocytes is generated. It can be described as a first injection, i.e. immune system initialization. In order not to lose possible answers on this step, we do not calculate affinity, leaving all lymphocytes in the system as is.

**Main step.** The algorithm of selection and cloning is used as the main one in the immune network. The main idea of algorithm is that the lymphocytes improve the measure of affinity to certain antigens. The degree of fitness of an organism is an affinity measure following the Darwinian Theory; the reproduction is carried out by means lymphocytes. Thus the realization of this algorithm by artificial immune network is possible.

**Step 1.** Let the initial set of lymphocytes be already created, the affinity measure is calculated. We get out l set of lymphocytes most strongly reacting. The mutation procedure is applied to choose a variety a set and probably the best approach to the answer. It is possible to generate missing lymphocytes by a random way. This step repeats after the active set of lymphocytes is ready. The decision is the criterion function represented by a lymphocyte from the current set with the greatest measure of affinity (figure 1).

Let’s see how in this artificial immune network this algorithm can be implemented.

1) After initialization, when the initial set of lymphocytes already formed, for each lymphocyte the measure of affinity is calculated.

2) Then, from the entire set of lymphocytes a certain number with the highest affinity is selected (most fitted in terms of selection and cloning algorithm).

3) Lymphocytes that are not selected in point 2 can not cope with the given problem, so they are removed from the model described.

4) In order to provide the selection diversity and receive the best possible approximation to the answer the hypermutation procedure to the selected in point 2 lymphocytes is applied. Hypermutation for the lymphocytes, representing an expression tree, can consist of the following actions: replacing the randomly selected action by another one, changing the numerical constant or variable, addition / removal of a sub tree of the randomly selected tree node.

5) In order to ensure the constant size of the population an operation of removal needs to be executed. To ensure a constant number of lymphocytes in the active set, one can either use only the preservation of the current set of best lymphocytes, along with their hypermutated descendants; or regenerate the missing lymphocytes randomly anew (“fresh blood” in the living body is the new lymphocytes produced by the bone marrow constantly).

6) When the active set of lymphocytes gains the needed number of cells, action in point 2 is repeated.

Stopping criteria is either an acceptable value of the objective function or an execution of the maximum number of iterations. The solution is a function representing by the lymphocyte from the current set with the greatest measure of affinity.

4. The description of the distributed version of artificial immune system

On each compute node the artificial immune system, described in the previous section, is functioning, that is, firstly a set of randomly generated lymphocytes is created, then operation of selection and mutation is conducted. However, each node has information on at least about one other functioning node. Therefore, once a certain number of iterations is defined (after the step of hypermutation of selected lymphocytes), then the given immune system requests and receives a set of best lymphocytes of another node. The resulting lymphocytes are involved in creation of a new active set of lymphocytes. Similarly, the given node when accessed with another will send to the last a set of the best lymphocytes. Thus, this distribution system allows all nodes to share their most successful results between each other. At the same time, if one or more
nodes of the system drop out, the work of the network will not stop, simply the other up and running nodes will be interviewed. For a node that just begins its function, when some nodes have already achieved some results, the work will not start from the very beginning. The "fresh" node will receive the results from already functioning ones and will start its work accordingly. The algorithm of functioning of the distributed network for two nodes is presented in figure 2.

The creation of the distributed version of artificial immune system allows carrying out expensive calculations quicker than several computing systems. This distributed system works in the heterogeneous environment consisting of various computing blocks. The computing block is a computer supporting a network stack of TCP/IP with the chosen programming language Python.

The architecture of interaction of computing blocks is the chosen interaction p2p model. The network is the computer network based on equality of participants. There are no allocated servers, and each node is both the client, and the server in such network. The concept of the distributed immune system has been chosen due to the reason that immune cells are distributed
within the whole organism. Based on the above mentioned we construct the following model (figure 3).

**Figure 3.** The algorithm of the distributed artificial immune system

5. Realization
The main program consists of the following parts [16]:
   1) Subsystem solution of the symbolic regression problem.
      1.1 The module of a simple artificial system contains the classes of lymphocytes along with the methods to simplify expressions, to compute affinity, and the algorithm of sequential implementation of the artificial immune system.
1.2 The module for the organization of the interaction between artificial immune systems in the network.
1.3 The module of implementation of a distributed artificial immune system.
1.4 The module of testing of the different functions of the subsystem, including the methods of computational experiment.

For the realization the Python language (version 3.3) has been selected [17]. This language has been chosen for the following reasons [17]:

- Cross-platform reason: during the creation of an application, running in a heterogeneous computing environment, the cross-platform factor is very important. Python allows you to create a program that does not require modification of the source code and recompiling even to work on a variety of platforms.
- The speed of development reason: Python is oriented on the increase of developer productivity and code readability. It also supports a variety of programming paradigms: structured, object-oriented, and functional [17].
- The standard library reason: includes a large number of useful functions and classes for various tasks such as networking, multithreading, process management.

During the creation process a system of version control Git has been utilized, with the source code stored on Github.

6. Result accuracy. Test examples, comparison with the existing methods

Let's compare the work of genetic programming algorithms with the offered the immune system.

Let's consider the functions and the convergence of methods described in [18], where authors show some experimental results of the genetic programming algorithm based on the Trees Addition Grammar (TAG). Trees Addition Grammar – this is a formal grammar, created by Aravind Joshi. This grammar summarizes the context-free grammar in such a way that the elementary unit in the rules of inference become not the individual characters, but trees. Thus, the grammar defines the rules of tree nodes replacement to subtrees. In their article authors compared the results of the new algorithm (TAG3P) with the results of a simple algorithm of genetic programming (GP) and genetic programming algorithm based on grammars (GGGP).

To conduct a test authors use the following functions:

\[
\begin{align*}
  x^2 + x; \\
  x^3 + x^2 + x; \\
  x^4 + x^3 + x^2 + x; \\
  \cos(2x).
\end{align*}
\]

For genetic programming algorithms [18] the authors use the following settings:

1) Terminal expression of the used expression trees: \(x\) – independent variable for the searched function – polynomial, \(x\), and real numbers for the function – the cosine.

2) Supported operators: binary \{+,−,∗,\}; unary: sin, cos, exp, log polynomials for functions, sin, for the function – the cosine.

3) Initial data: the values of 20 randomly selected points in the interval \([-1, 1]\) for polynomials and 20 randomly selected points in the interval \([0, 2]\) for the cosine.

4) The objective function: the amount of errors in the 20 specified points.

5) Genetic operation: tournament selection, a one-point crossover and replacement of genes – for algorithm TAG3P. Tournament selection, standard crossover and mutation for GP and GGGP algorithms.

6) Parameters of genetic algorithms: the probability of crossover for GP, GGGP and TAG3P equals to 0.9. The probability mutation for GP and GGGP equals to 0.1. The probability of
gene replacement algorithm TAG3P equals to 0.01 for the functions of polynomials, and 0.04 for the cosine. The number of individuals participating in the tournament equals 3. Maximum number of generations is 30 for the functions and polynomials, and 200 for the cosine. The population size is 500 individuals.

7) The criterion of the successful work of the algorithm: in all 20 points achieved the accuracy of 0.01.

8) The goal is to find a function of a single variable that fits into the initial data the best (calculating values of 20 points).

For each algorithm and for each function the algorithm run 50 times. The obtained results are shown in the table below (table 1) [18].

### Table 1. Results of the genetic programming algorithms.

| Function       | GP  | GGGP | TAG3GP |
|----------------|-----|------|--------|
| $x^2 + x$      | 47 (94 %) | 46 (92 %) | 50 (100 %) |
| $x^3 + x^2 + x$ | 30 (60 %) | 32 (64 %) | 50 (100 %) |
| $x^4 + x^3 + x^2 + x$ | 21 (42 %) | 24 (48 %) | 48 (96 %) |
| $\cos(2x)$    | 0 (0 %) | 10 (20 %) | 18 (36 %) |

Now lets consider the characteristics of the used immune system:

1) Permitted functions: {\+, -, \*, \/, \sin, \cos}.

2) Terminal elements: variable \(X\) and the real numbers (unlike in genetic algorithms).

3) The maximum height of the tree: 4 (can be changed to more complex functions).

4) The number of lymphocytes in the system is 250 (2 times less than the number of individuals in the genetic algorithms).

5) Maximum number of steps is 150 (steps in the immune system consist only of a simple selection and mutation, so their number can be increased).

6) The criterion that a solution is found (stopping criterion) is when the solution obtained with the accuracy 0.01. In contrast to the criteria of successful work of genetic programming algorithms, in the immune system successfully stop means that the objective function has a value of less than 0.01.

Let’s recall that as an objective function the following function is used:

\[
\sqrt{\sum_{i=1}^{r} (f(x_i) - y_i)^2},
\]

where \(x_i \in [-5, 5]\) – free variable; \{\(y_1, \ldots, y_r\)\} – specified set of values of the search function at the corresponding points \(x_i\) \((r = \text{amount of points})\); \(f(x_i), i = 1, r\) – multiple values of function represented by this lymphocyte, at the corresponding points \(x_i\).

As a result we obtained that the immune system uses a more stringent criterion for the successful solution of the problem, so we extend the segment, from which the values of the independent variable are taken, to \([-5, 5]\). On the segment with small values the immune system, already at the first steps, finds a reasonably well approximating function given with an accuracy of less than 0.5, so we extend the segment.

In conclusion we found that artificial immune system in all cases was able to solve the problem better than the 2 algorithm of genetic programming (GP and GGGP), showing comparable results with TAG3GP algorithm – better for the cosine, and a little worse for polynomials.
Based on the above described examples and comparison with existing methods one can draw a conclusion about the competitiveness of artificial immune systems application in solving the problem of symbolic regression.

7. Conclusion
In this paper the model of the distributed immune system was introduced. Also a structure of the system algorithm and its parallel realization were presented. This model allows to carry out expensive calculations quicker. This distributed system works in the heterogeneous computing systems. In this case the computing block could be any modern computer.

The model of p2p (peer to peer) interaction is considered. The network is a number of computers with equal rights; any node is client and server. The Python language was selected for realization.

References
[1] Koza J R 1998 Genetic Programming (Cambridge: The MIT Press)
[2] Hart E and Timmis J 2008 Applied Soft Computing \textbf{8} 191–201
[3] Dasgupta D, Yu S and Nin F 2011 Applied Soft Computing \textbf{11} 1574–1587
[4] Bersini H 1999 Artificial Immune Systems and Their Applications 22–44
[5] Uy N Q, Hoai N X, O’Neill M, McKay R I and Galván-López E 2011 Genetic Programming and Evolvable Machines \textbf{12} 91–119
[6] Hoai N X, McKay R I and Essam D C 2002 Evolutionary Computation, 2002. CEC’02. Proceedings of the 2002 Congress on \textbf{2} 1326–1331
[7] Watkins A, Timmis J and Boggess L 2004 Genetic Programming and Evolvable Machines \textbf{5} 291–317
[8] Hart E and Timmis J 2008 Applied Soft Computing \textbf{8} 191–201
[9] Breiman L 2001 Machine Learning \textbf{45} 5–32
[10] Astakhova I F, Ushakov S A and Hitkova J V 2015 (Rome, Italy: Proceeding of the 15-th International Conference on Evolutionary computing (EC’15), Proceeding of the 6-th European Conference of Computer Science (ECC’15)) pp 127–131
[11] Johnson C G, Ryan C, Soule T and Keijzer M 2003 EuroGP 2003. \textit{LNCS} \textbf{2610} 345–353
[12] Farmer J D, Packard N and Perelson A 1986 \textit{Physica D} \textbf{22} 187–204
[13] Garrett S 2005 \textit{Evolutionary Computation} \textbf{13} 145–178
[14] Cutello V and Nicosia G 2002 Lecture Notes in Computer Science, Springer \textbf{2527} 361–370
[15] Chio C D, Agapions A and Cagnoni S 2012 Lecture Notes in Computer Science, Springer \textbf{7248} 477–486
[16] URL https://github.com/StanislavUshakov/ArtificialImmuneSystem
[17] URL https://www.python.org/
[18] Hoai N X, McKay R I and Essam D 2002 Evolutionary Computation, CEC ’02. Proceedings of the 2002 Congress \textbf{2} 1326–1331