# An Inductive Immune Algorithm Based on the Cooperation Principles

Bidjuk P.I[1]., Bardachov J.N.[2], Litvinenko V.I.[2,3], Fefelov A.A.[2],Hodakovskij A.V.[3]
[1]*National Technical University of Ukraine "KPI"*
*03056,Kiev, Peremogy 37*
[2] *Kherson National Technical University*
*73008 Kherson, Berislavske Shosse 24*
[3]*Kherson State Marine Institute*
*73000, Kherson, Ushakov Prospect 17*
peter@bidyuk.carrier.kiev.ua, kstu@tlc.kherson.ua, johndoe2004@mail.ru, fao1976@mail.ru, warlock@mail.ru

**Abstract.** *An approach to solve an approximation problem by means of immune algorithm that is based on the principle of cooperation of population antibodies is offered. The formal description of structure of an antibody and ways of their association within the limits of a population in the computer network functioning as a unit is given. The way of antibodies estimation, that are considered as elements of a network, is proposed. Description of the training algorithm based on a principle of clonal selection is presented. The basic phases of functioning of the algorithm are considered, such as*: *growth of a network, mutation of cells, and network compression.*

### Keywords

immune network, clonal algorithm, approximation, forecasting, cooperative immune algorithm

## 1. Introduction

The basic feature of all existing for today population algorithms is the principle of competition of individuals inside of the population. The given principle is implemented due to selection of the best individuals, granting them this way an opportunity to be reproduced and, as a consequence, to pass into new generation. Competitive process, however, assumes necessity of coding within the limits of one individual solution of a problem entirely. Thus, the population consists of a set of alternative solutions which in the process of functioning of the algorithm undergo the changes directed to improvement of their quality. Immune algorithms are population algorithms [1]. They develop a population of individuals named as antibodies that are capable to distinguish intrusion of alien bodies named as antigens. One of possible applications of the immune algorithms are the problems of approximation in which each antibody of a population represents a full mathematical description of model of approximated data. During a competition and change of antibodies it is formed one or several best models that become a final solution to the approximation problem. To solve the same kind of problem in the given work it is offered to use a principle of cooperation of antibodies instead of their competition. In this case each antibody of a population is only a part of the solution or in a case of approximation problem – a part of a model. Within the limits of a population of antibodies they interact with each other in a certain way (cooperate) and form a structure, capable to solve the problems at a system level [2], i.e. at the level of all population instead of level of a separate individual.

## 2. Problem statement

The problem of model identification of nonlinear structure of the following kind is considered:

$$y = f(x_1, x_2, ..., x_n), \quad (1)$$

for which dependence between inputs $x_i$ and an output $y$ is presented in the form of the table of given experiments *T*:

$$T = \bigcup_{i=1}^{k} \left( x_{i1}, x_{i2}, ..., x_{in}, y_i \right), \quad (2)$$

where $k$ is a number of lines in the table. Both input and output stochastic processes can have arbitrary distribution, and exhibit nonstationary behavior. It means that problem statement is given in the most general form.

# 3. Representation

Relation (1) generally represents mathematical expression which can be written in the form of some formula. For example, we shall admit, that our object has three inputs $\left( x_1, x_2, x_3 \right)$, and one output. We shall also admit that dependence between inputs of the object and its output can be described by means of the expression:

$$y = a_{12} \cdot (a_1 x_1 + a_2 x_2) \cdot a_3 x_3, \quad (3)$$

where $a_1, a_2, a_3, a_{12}$ are some constants which are carrying out function of factors. In this case the expression (3) can be represented in the form of the column given in Fig. 1.

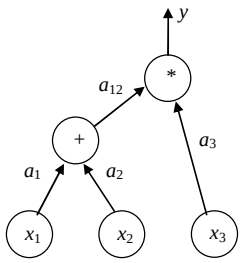


**Fig. 1.** Representation of mathematical expression in the form of the column

In terms of the theory of evolutionary algorithms such columns are referred to as genetic program. It contains vertices of two types:

- terminal vertex – vertex which has no arches entering into it; such tops represent variable problems;
- functional vertex – vertex which has both input, and processing arches; functional vertices contain signs of mathematical operations and functions.

Each vertex can have a set of processing arches. The input arches can have only functional tops. The number of arches entering into vertex depends on the quantity of arguments of function which is represented with the vertex. As the majority of mathematical operations and functions are monadic or binary the most part of vertices will have one or two entering arches. Unlike classical representation of a genetic program, in the given work all column arches are weighed. One of the basic properties the column is that it does not contain cycles and can be always represented in the form of a tree. Let's consider one more example of mathematical expression:

$$y = a_{123}\left(a_{12} \cdot (a_1 x_1 + a_2 x_2) \cdot a_3 x_3\right) + a'_{12}\left(a'_1 x_1 \cdot a'_2 x_2\right) \quad (4)$$

The genetic program corresponding to expression (4) is shown in figure 2.

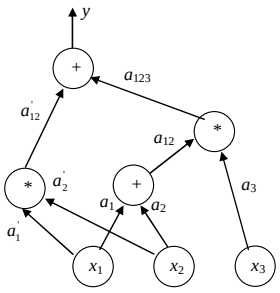


**Fig. 2.** Columns of mathematical expression (genetic program), corresponding to expression (4)

# 4. Coding of antibodies

Development of artificial immune systems starts with definition of the search space or space of forms. Each form is identified with an antibody of immune system and represents a line coding one possible solution of a problem. As a space of forms could be used binary, integer, material, symbolic space or their combinations. The method offered in given work, unlike classical approaches, uses the population coding of genetic programs instead of individual. Individual coding assumes representation of all genetic program in the form of a line within the limits of one antibody of a population. In population coding each antibody is only a part of the genetic program, i.e. the column or one unit of a tree of mathematical expression is coding one vertex. The set of terminal vertices of a column is not exposed to any changes during solution of a problem. Hence, the antibodies, coding functional vertices should make a population of immune algorithm only. Within the limits of the given work we shall be limited to consideration of monadic and binary mathematical operations and functions. In this case the line of an antibody can be presented as shown in Fig. 3.

| Code of function | Code of 1-st unit-descendant | Code of 2-nd unit-descendant | Weight of 1-st unit-descendant | Weight of 2-nd unit-descendant |
|---|---|---|---|---|

**Fig. 4.** The offered structure of an individual of immune algorithm population

It is apparent from the figure, that antibody has mixed (integer and material) coding. The code of function and codes of units are represented by integers, and weights are real. The code of function is identified by its number or an index in a file of functions (functional set). The functional set *(F)* can contain any quantity of mathematical operations, functions and polynomials, i.e $F = \{+, -, *, /, \sin, \cos, \tan, \ln, poly1, poly2, ...\}$. Here as $poly1, poly2, ...$ we use, for example, Kolmogorov-Gabor polynomials: $f_1(x_1, x_2) = c + a_1x_1 + a_2x_1x_2$ $f_2(x_1, x_2) = c + a_1x_2 + a_2x_1^2$ , etc. All vertices of the genetic program column are numbered. Numbers of vertices are assigned to arcs directed to the given vertex, and are stored in section of codes of descendants units of a given vertex. Owing to that all vertices could contain any quantity of processing arches at any connection of the given vertex to the column we shall not receive syntactically incorrect mathematical expression, even in the case of when codes of both units of descendants will appear equal. This important property the column during training allows us to provide a maximum freedom of evolution of a structure.

# 5. Calculation of affinity

Any immune algorithm assumes presence of a population of antigens *(AG)* which will be distinguished by a population of antibodies *(AB)*. In problems of a system model identification as a population of antigens is used the set (table) of experimental data lines *T*: $Ag_i = t_i, Ag_i \in AG, t_i \in T, i = \overline{1, k}$ . There is also an opportunity proposed in [4], that is in splitting of the experimental data table into subsets of lines as $Ag_i = T_i, T_i \subseteq T$ , and generally $|Ag_i| \neq |Ag_j|$, $Ag_i \cap Ag_j \neq \varnothing$ at $i \neq j$ . The estimation *of i-th* genetic program and *j-th* antigene is calculated as Euclidean distance. In given work as calculated affinity value we use a

degree of similarity instead of degree of complementary individuals. Therefore we introduce additional function of affinity (to keep a target direction of maximization of affinity) in the form of:

$$f_{aff} = \frac{1}{1+D_E}. \quad (6)$$

As in the given work each antibody represents only a part of genetic program its affinity will be calculated on the basis of an estimation of the subgraph, formed by current vertex and its all descendants. Thus, values of function $f_{aff}$ change in an interval from 0 up to 1, i.e $f_{aff} : \Re^+ \to [0,1]$. For management of sensitivity of antibodies the threshold of affinity ($\varepsilon$) is introduced. The fact of recognition considers an antibody $Ab_i$, $Ab_i \in AB$ of an antigen $Ag_j$, for which the value of affinity function is $f_{aff} \geq \varepsilon$. Hence, the less is the value of the affinity threshold, the higher will be robustness of a system. The quantity of the antigens, distinguished by an antibody $Ab_i$, refers to as concentration of antigens and is designated as $\nu_i^{Ag}$. For calculating the concentration we shall define function of linkage of an antibody to antigen as:

$$b : [0,1] \times [0,1] \to \{0,1\}. \quad (7)$$

This function can accept only two values: 1 – linkage has occurred (an antigen is distinguished by an antibody) and 0 – linkage has not occurred (an antigen is not distinguished by an antibody). Using values of function $f_{aff}$ and the threshold $\varepsilon$, it is possible to represent function *b* in the form of the following parity:

$$b = \begin{cases} 0, & if \ f_{aff} < \varepsilon; \\ 1, & if \ f_{aff} \geq \varepsilon. \end{cases} \quad (8)$$

Then value of concentration of an antigen for an antibody $Ab_i$ can be calculated so:

$$\nu_i^{Ag} = \sum_{j=1}^{k} b_{ij}. \quad (9)$$

where $b_{ij}$ - value of function of linkage of an antibody $Ab_i$ with an antigen $Ag_j$. Stimulated the cell, at which is considered value $\nu_i^{Ag} > 0$.

Let's consider the fragment of a population represented in Fig. 5. Using (5) – (9) concentration $\nu_3^{Ag}$ can be presented parities as some composition of concentration $\nu_1^{Ag}$ and $\nu_2^{Ag}$, i.e. $\nu_3^{Ag} = \nu_1^{Ag} \circ \nu_2^{Ag}$. In this case it is possible to allocate the following two cases of a parity of these three sizes. Case 1: $\nu_3^{Ag} > \max(\nu_1^{Ag}, \nu_2^{Ag})$ – an antibody $Ab_3$ has greater concentration of antigens in comparison with antibodies $Ab_1$ and $Ab_2$. Hence, from the point of view of the genetic program, introduction of unit $Ab_3$ improves approximation of dependence, approaching us to the problem solution. In this case the cell $Ab_3$ is stimulated and concentration of an antigen for it remains equal $\nu_3^{Ag}$. A variant 2 $\nu_3^{Ag} \leq max(\nu_1^{Ag}, \nu_2^{Ag})$. Here introduction of unit $Ab_3$ worsens or does not change approximation of expression from what follows, that the cell $Ab_3$ is not stimulated and its value $\nu_3^{Ag}$ is equated to 0. The algorithm of training uses the information on stimulated and not stimulated cells to increase or reduce the size of repertoir of antibodies.

## 6. Algorithm of training

Generally at the beginning of training process the presence of only one antibody in a population of antibodies is supposed. During training the population of antibodies will be structured in the form of the column, similar to the column presented in figure 2. In a context of immune algorithms we shall name the given structure a functional network of antibodies (ΦСА). As the basic properties of the given network it is possible to allocate the following: 1 – the growth of a network which is based a principle of clonal selection; 2 – the compression of a network based on destruction of not stimulated cells (apoptosis); 3 – evolution of adjustments and structures of the network, based on estimation mechanisms and somatic hypermutation. In general view the training algorithm can be described as follows.

**Step 1**. ***Initialization.*** Creation of an initial population of antibodies *AB*. In the given work the initial population consists of one antibody, but initialization of a population of any predetermined size is possible.

**Step 2.** ***Infecting (presence of antigens).*** For each antibody $Ab_i \in AB$ on the basis of expression (6) is to be calculated the concentration of an antigens, $\nu_i^{Ag}$ .

**Step 3.** ***Selection and cloning***. Choose an antibody with the greatest value of concentration (the most stimulated cell), and clone the chosen antibody. Forming of both one, and several clones is possible.

**Step 4**. ***Maturing of affinity.*** Subject to mutations all clones of the chosen cell with the intensity that is inversely proportional to their values of concentration, $\nu_i^{Ag}$ . During a mutation change a code of function and codes of the first and second descendants (evolution of structure of a network), and also the change of weights of the first and the second descendants (evolution of adjustments of a network) is possible.

**Step 5**. ***Repeatedly calculate concentration*** of antibodies of a network, similarly to a step 2.

**Step 6.** ***Compression of a network***. Remove all cells of the network, which have value of concentration, $\nu_i^{Ag}$ , that is less or equal to some set threshold, $\nu_{\min}^{Ag}$ , (removal of not stimulated cells).

**Step 7**. Go to step 2, if the break condition is not satisfied.

## Growth of a network

This process represents a choice and reproduction of the most stimulated cell of a population according to the principle of clonal selection. The choice of a cell for cloning occurs according to values of concentration of the antigens, calculated for all cells of a network. The cell with the highest concentration gets out for cloning.

## Hypermutation of cells

The mutation plays an important role in forming of immune answer of system on influence of an antigen. Owing to the mechanism of a mutation there is an adaptation of structure of antibodies and, as consequence, increase in affinity of a population. In the given work intensity of a mutation depends on values of concentration, $\nu^{Ag}$ , of antibodies of a network. The higher is the value of $\nu^{Ag}$ , the less is intensity of a mutation. Any part of a line of an antibody can be subjected to a mutation. Intensity $\beta$ is understood as a quantity of elementary influences of the operator of a mutation on antibody *Ab*. Elementary influence is made under the scheme of the one-dot mutation offered in [Holland]. As the operator can influence any part of an antibody it can update not only weight characteristics of arches column AIS, but also mostly the structure of AIS. One more important fact is that in the given example the network, owing to a mutation, "has got rid" of the second output, having formed uniform structure. To avoid forming of a significant amount of outputs of a network, in an offered method the restriction is proposed according to which any functional vertex of a network can form a link only with terminal vertices or other functional vertices which do not have processing arches.

## Compression of a network

Compression of a network means the process of destruction of not stimulated cells, leading to reduction of the network size. Consider the process of network compression on example (fig. 10). In this case the cell $Ab_2$ has concentration of antigens, $\nu_2^{Ag} = 0$ , i.e. it is not stimulated.

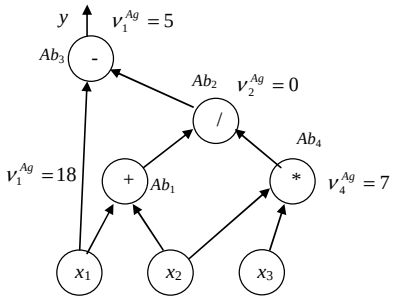


**Fig. 10.** The state of a network before compression. The antibody $Ab_2$ is not stimulated and should be removed from a network

Removal of the vertex $Ab_2$ leads to forming of syntactically incorrect expression since an antibody $Ab_3$ does not receive the second argument for binary mathematical action – subtraction.

# 7. Conclusions

In the paper the approach to solution of the approximation problem by means of immune algorithm, based on the principle of cooperation of antibodies of a population, is offered. As a result we get a formal description of antibodies structure and ways of their association within the limits of a population in the functioning computer network. The way of estimation of antibodies as elements of a network is also considered. A description of the training algorithm based on a principle of clonal selection is resulted. Separately examples of basic phases of the algorithm are considered: growth of a network, a mutation of cells, compression of a network. Growth of a network is carried out due to selection and cloning of the best antibodies; the mutation of cells changes structure and adjustments of a network, and process of compression is provided by removal of not stimulated antibodies and results in reduction of the network size. This process plays exclusively important role, as owing to it the algorithm aspires to create solutions of minimal size. However removal even of one cell can essentially affect functioning of a network as a whole. Depending on position of the deleted cell, the given procedure can lead to strong redistribution of values of concentration of antigens among the remained antibodies. An analysis of the algorithm allows to tell, that it tends to delete only cells of top level, i.e. the networks closest to outputs and, hence, will not cause global reorganizations of structure. On the other hand it is possible to state with confidence, that process of compression allows a network to grow only aside increases in values of concentration of antigens, and the process of search initiated by clonal selection and a mutation, provides adding to a network of new cells consistently approaching a network to solution of a problem As the further research in this area it is possible to allocate to following directions:

- implementation of some alternative ways of estimation of antibodies as elements of the uniform computing system;
- further improvement of growth procedures and compression of a network;
- search and elimination of superfluous parts of a network with the purpose of simplification of its structure.

# References

[1.] Leandro Nunes de Castro, Fernando J. Von Zuben. Artificial immune systems: part 1 - basic theory and applications. TR - DCA 01/99 1999.
[2.] Castro, L. N., Von Zuben, F. J., Getulio A. De Deus Jr. The Construction of a Boolean Competitive Neural Network Using Ideas from Immunology, Neurocomputing, 50C, pp. 51-85, 2003.
[3.] J. R. Koza. Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge, MA: The MIT Press, 1992.
[4.] Nikolaev N., Hitoshi I., Inductive Genetic Programming of Polynomial Learning Networks, X.Yao (Ed.), ECNN-2000, IEEE Press, pp. 158-167.
[5.] Ferreira C. Gene Expression Programming in Problem Solving, WSC6 tutorial, 2001.
[6.] Holland, J.H. Adaptation in Natural and Artificial Systems, Ann Arbor, MI: The University of Michigan Press. 2nd ed. (1992) Boston, MA: MIT Press.