UDC 004.89

Korablev N. M.[1], Ivaschenko G. S.[2]

[1]Dr.Sc., Professor, Professor of the Department of Electronic Computers, Kharkiv National University or Radio Electronics,Ukraine
[2]Postgraduate student of the Department of Electronic Computers, Kharkiv National University or Radio Electronics, Ukraine

PARALLEL IMMUNE ALGORITHM OF SHORT-TERM FORECASTING BASED ON MODEL OF CLONAL SELECTION

The paper studies ways of parallelization of hybrid immune algorithm of short-term forecasting of time series built on the basis of clonal selection model that uses case-based reasoning and the simplest methods of forecasting. There has been analysis performed of two variants of parallelization having different procedure of messaging between the computational nodes. To implement proposed algorithm used MPI.NET technology for messaging systems. To optimize individual computational nodes operations, TPL library for shared memory systems is used. The work presents results of experimental investigations demonstrating efficiency of the proposed approach.

Keywords: forecasting, time series, artificial immune systems, clonal selection model, antibody, antigen, affinity, cloning, mutation.

NOMENCLATURE

AIS is an artificial immune systems;
CBR is a case based reasoning;
MPI is a message passing interface;
TPL is a task parallel library;
N is a number of known values of forecasting time series;
N_m is a number of forecasting methods, which are used in current model;
N_g is a number of iterations during learning of AIS;
k is a length of forecasting horizon;
S is a number of external factors;
L is a length of antibody part which takes part on the affinity determine;
Ab is a population of antibodies;
Ab^C is a population of clones;
Ab^M is a population of memory cell;
Ag is a population of antigens;
n is a number of antibodies in Ab;
n_m is a size of population of memory cell;
n' is a number of cells selected for cloning and mutation;
n_c is a number of clones made by one antibody;
D_{Ab-Ag} is a matrix of *Ab*-*Ag* affinity;
D_{Ab-Ab} is a matrix of *Ab*-*Ab* affinity;
σ_d is a threshold coefficient of cell stimulation;
σ_S is a threshold coefficient of compression;
σ_{age} is a threshold age of antibodies;
N_{cn} is a number of computational nodes;
T is a duration of communication operations;
V_{ag} is amount of data to transmit one antigen;
V_{ab} is amount of data to transmit one antibody;
α is a latency of a data network (Hockney model parameter);
β is a bandwidth of a data network (Hockney model parameter).

INTRODUCTION

Time series forecasting allows to solve a problem of determination of the future condition of different systems on the basis of analysis of already available historical data. The task of the short-term forecasting is essential to technical and economic systems since the estimation of the future values of a series is used for efficient decision making. Despite the fact that the development of hardware and software provides with more powerful computing platforms due to which it is possible to implement complex algorithms of forecasting, the tasks of economic and technical management makes more severe requirements to the accuracy of forecasting [1]. Taking into account of external factors represented in the form of the related time series allows to improve the accuracy of forecasting significantly. However, a prior list of factors potentially influencing the predicted value is often redundant. That is why a problem of external factors choosing taken into the consideration during the forecasting continues to be relevant [6].

One actively develops approaches to the forecasting of time series on the basis of artificial intelligence techniques such as artificial neural networks and AIS [3] that can be integrated with various approaches and are notable for easy parallelization and abilities to adapt. There are different models based on the principles of immune system: model of clonal selection, model of immune network etc., that can be used to solve forecasting problems [2, 4–6]. These models can use different forecasting techniques that allows to timely take into consideration changes in the basic structure of time series and to compensate for shortcomings of one approaches by means of advantages of the others [2]. Simplicity of the converting to parallel is one of the immune algorithm advantages.

At the present time companies have accumulated a considerable volume of historical data that leads to the significant growth of volumes of input information for forecasting problem. Thus, in case of growing volume of

DOI 10.15588/1607-3274-2014-2-11

input data there is a necessity to solve tasks of short-term forecasting under the conditions of limited computational resources and time limit. This is why a relevant way to solve this problem is a parallel programming for multi-core and multiple-processor cluster systems. However, while writing parallel programs algorithms of solving even the simplest problems ceases to be trivial [7].

The object of the research is developing a parallel immune algorithm of short-term forecasting of time series based on the clonal selection model and investigation of its efficiency.

1 PROBLEM STATEMENT

One set time series $Z(t) = z_1, z_2, ..., z_N$ lengths N and S external factors represented in the form of time series $X_1(t), X_2(t), ..., X_S(t)$, the values of which are obtained in the time points as follows $t_1, t_2, .., t_N$. It is required to determine future values of the series $Z(t) = z_{N+1}, z_{N+2}, ..., z_{N+k}$, where k is a length of forecasting horizon.

Application of approaches based on different models of AIS allows to achieve good results in solving the problem of short-term forecasting [2, 4–6], but their using characterized by a high computational complexity. One of the ways of efficient organization of computational process may be to use of distributed computing on the multiprocessing systems, for example, using MPI technologies. It is required to perform algorithm decomposition and to propose ways of distribution of problems among processors, thereafter to compare the efficiency of the methods proposed by means of evaluation of communication complexity according to Hockney's model [7].

The greatest influence on timing characteristics of immune algorithm has the size of population of antibodies. It is required to determine dependence of AIS learning time on the number of antibodies, thereafter to conduct similar experiments for multiprocessor and multi-core architecture.

2 REVIEW OF THE LITERATURE

Proposed in [5] immune algorithm based on the immune network model that uses CBR is successfully applied to solve problems of short-term forecasting, but it demonstrates a poor result for short-memory time series that is defined by the peculiarities of CBR. Partially this problem is solved by the approach [2], based on the application of clonal selection model using heterogeneous antibodies built on the basis of CBR and the simplest forecasting techniques. This approach uses a segmentation of time series and selection for each segment the most efficient forecasting technique. This allows achieving the higher efficiency rather than the application of every predictor individually of the given set for forecasting of the entire basic series. The further development of this approach [6] allows not only to make a selection of the actual external factors, but also it successfully works with the distorted time series, however, this requires a significant increase in the number of antibodies in the population.

Time series forecasting using proposed in [6] immune algorithm is a computationally capacious procedure since, at first, it is required to train a system on a large training set. To solve this problem it is reasonable to use tools for organization of the distributed computing [7], such as MPI library, and TPL [8]. TPL allows significantly facilitating a writing of a maintainable code that automatically uses the advantages of multi-core systems without changing the application design [8]. This technology will be efficient, if the algorithm includes many cycles that use shared arrays of data.

In the paper presented here the approach proposed in [6] obtained further development using the technologies of parallel programming for multi-core and multiple-processor cluster systems.

3 MATERIALS AND METHODS

In this study the model of clonal selection [2, 4, 6] was chosen for parallelization. This model is based on searching of antibodies (decision options), which most fully conform to an antigen (original problem), basing on the knowledge of the affinity (proximity measure between antibody and antigen).

It is possible to present the formal way of AIS model to forecast problem solving [2, 6] in following manner:

$$AIS\_CS = \left\langle Ab, n', n_c, Ab^C, Ab^M, Ag, D_{Ab-Ag}, \sigma_d, D_{Ab-Ab}, \sigma_s, \sigma_{age} \right\rangle. \quad (1)$$

Antibody acts like precedent and consists of two parts. The structure of the first part is similar to one of antigen (but it includes sample of one outside factor only), and it represents a set of parameters, describing the problem statement (in our case that are samples of series with known values, including missing values), and it is used to define affinity value. The second part, the length of which is equal to the forecast horizon does not affect to the calculated value of affinity of the antibody and describes the proposed prediction for that sample values of the time series, which is the first part of antibody.

Antigen is combination of values of forecasting time series and attendant series going ahead directly before forecasting values. In that case, forecast problem comes down to a problem of antibody search with the max affinity value, thus, it is search of sample which matches the most to values at the last part of the time series.

In order to use advantages of the way presented at [6], it is required usage of the big size population of antibodies. Overall number of antibodies included in population without missing values of predicted time series can be defined by following way:

$$n = \sum_{i=L_{\min}}^{L_{\max}} \left(\left(N - \left(L_i + k \right) + 1 \right) N_m S \right). \quad (2)$$

Using of different possible values of antibody length allows to configure the AIS to forecasting time series [5].

This work represents two methods of immune algorithm parallelization based on island model of the genetic algorithm [9]. Using data parallelism in the task consists in uniform

(except for the control node) distribution of antibodies population among computational nodes and follow-up independently of evolutionary process by every computation node. It helps to use sizeable population of antibodies when using serial version of immune algorithm will exact intolerably much of work time.

General concept for these two parallel realizations of the immune algorithm is using of control process (rank of which is equal 0), which distributes the starting data and forms result after end of work of N_{cn} computational nodes (with rank more than 0). Control node doesn't have part of antibodies population for cultivation. Size of population on the computational nodes is defined as

$$n_1 = \frac{n}{N_{cn} - 1}. \quad (3)$$

First way of parallelization of the immune algorithm is:

1. Each of the computational nodes generates the unique antibodies populations which have size n_1.

2. The control node delivers the antigen to computational nodes.

3. Each of the computational nodes selects the antibodies with the most affinity and sends to control node the clone subpopulation which have size $n_c n'$.

4. The control node compares the variants of forecast, selects the antibody with the biggest affinity value and forms the result on this iteration.

5. If there is a real value of predicted value, the control node compares the forecast variants of selected antibodies and real value, performs mutation of antibodies of clone population and corrects the coefficients which determine the influence of antibodies type and external factors on the antibodies affinity [6].

6. The control node resolves about inculcation or deletion of clones and being formed the memory cell population Ab^M for each node.

7. The memory cell population and changed parameters for calculating of the affinity are sent to the initial computational nodes.

8. Steps 2–7 are repeated for each antigen from the training set to achieve one of the stop criteria – carrying out a predetermined number of antibodies generations or reducing forecast error up to the user-specified value.

After the end of the training phase the order of interaction between nodes remains the same. This method of parallelization ensures the saving of unique populations of antibodies on each computational node, having shared between nodes only operations affinity determination and selection of antibodies.

To estimate the time of data transmission between compute nodes, the model proposed by Hockney is used [7]. The volume of data transferred on different iterations of the algorithm may vary significantly. After forming of the population at each iteration control node sends one antigen to each computational node, receives from a population of clones, and returns back to a population of memory cells, whose size is $n_m \in [0, n_c n']$. The volume of control data can be ignored. Then the duration of communication operations in the learning process of AIS for the first version of the parallel algorithm can be expressed as

$$T = (N_{cn} - 1)\left(N_g \left(\alpha + \frac{V_{ag} + V_{ab} n_c n' + V_{ab} n_m}{\beta} \right) \right). \quad (4)$$

Advantages of this approach are:

– unique antibodies in different populations at computational nodes, each node has an independent population of memory cells;

– supervision of algorithm parameters for each node at each iteration.

However, control of process in populations on computational nodes leads to considerable network load, since it occurs a transfer of large amounts of data on each iteration of the algorithm. In order to reduce the communication operations in the second version of the parallel immune algorithm is proposed distribution of the learning process of AIS and population control for individual computational nodes, without the exchange of intermediate results to the control node:

1. On each computational node it forms a unique population of antibodies which size is n_1.

2. The control node carry out the delivering of training set (antigen set and actual values of forecasting value) to all computational nodes. Every computational node performs step 3–7.

3. Selection and cloning of antibody with the highest affinity for the antigen.

4. Mutation of antibody clones from the population size $n_c n'$.

5. Correction of coefficients that determine the impact of realized in the antibody forecasting method and represented the external factor on antibody affinity.

6. Introducing or removing clones from the population regardless of the population of antibodies on other computational nodes.

7. Generates a population of memory cells Ab^M.

8. Steps 3–7 are repeated for each antigen from the training set to achieve one of the stop criteria.

After completion of the training phase the interaction between the control and the computational nodes is similar to steps 2–7 of the first variant of the algorithm. Computational nodes, having completed the work, send the best result to control node that determines the best individual from the sent options and generates the result of immune algorithm and adjusts the selection parameters on the computational nodes.

The main advantage of this approach is low network load in the learning process – the transfer of data is limited by the teaching selection, consisting of m antigen:

$$T = (N_{cn} - 1)\left(\alpha + \frac{m V_{ag}}{\beta} \right). \quad (5)$$

The disadvantage of this approach is the possible occurrence of the same antibodies in populations on different computational nodes during the learning process of AIS. Using of suppression will require additional data exchanges between nodes. In addition, there is an irregular loading of nodes at the end of the learning phase: when the best affinity is achieved on one process, there is a standing time, waiting the completion of the rest.

Since the proposed ways of parallelization of the immune algorithm differ mainly in the amount of information exchanges in the learning process of AIS, and further exchange of data is the same, then for the comparison of the approaches is sufficient to determine the duration of data transfer operations. Comparison of the obtained expressions 4–5 shows that the second developed method of parallel computation has considerably lower communication overheads and allows a better scalability as the number of processors increased. Therefore, further studies are carried out with the use of a second approach of the parallelization of the immune algorithm.

4 EXPERIMENTS

To check the efficiency of the parallel algorithm one chose a series of daily values of temperature (over 20000 values) that allows to create an extended training set. For computational experiment in this paper was used cluster, built on the basis of Microsoft Windows Compute Cluster Server 2003 (MS CCS) which supports the MPI.NET library. MPI.NET is an implementation of the MPI library for the platform .NET. This system provides safe and scalable management of the cluster resources, task scheduling, and presents a message passing interface for distributed programming [7]. Before the analysis the parallel version of the immune algorithm is required to determine the maximum size of the population of antibodies, for which it is advisable to use considered in [6], the immune algorithm. Obtained by computational experiment population size is then used in the determination of acceleration for a parallel version of the algorithm.

During the computational experiment was carried out sequential start of parallel algorithm with the same stopping criterion and maximum (for the serial version) population size for different (1–20) number of available computational nodes in the cluster.

For the analysis of efficiency of code optimization for multi-core architecture (used 2-core processor model Intel Pentium D, 3000 MHz) has been made a series of launches of the sequential version of the program adapted to using technology TPL, with different size populations of antibodies.

5 RESULTS

The tab. 1 shows the time of execution of one iteration of the algorithm during the AIS learning depending on the size of the antibodies population.

Fig. 1 illustrates dependence of training time of time series forecasting module on the involved number of processors showing the acceleration close to linear one. When decreasing a number of antibodies in the populations on particular computational nodes a superlinear acceleration is potentially possible.

In the immune algorithm of time series forecasting there are cycles (in particular, determination of affine properties in the populations antibodies and clones) for which a parallel processing is possible, in particular, affinity of every antibody in the population calculates without regard for another ones. As a result of the immune algorithm application adjusted for use of TPL technology an average acceleration has reached 7,5 % (tab. 2).

Fig. 2 illustrates a difference between times of training of the algorithm versions sequential and adjusted for TPL for Intel Pentium D, 3000 MHz dual core processor.

6 DISCUSSION

The results quoted in [6] of the short-term forecasting of time series represented in M3-Competition [10], and distorted series of daily average temperature confirm the effectiveness of the proposed immune algorithm for the short-term

Table 1 – The dependence of the execution time of the algorithm on the number of antibodies in the population for the serial immune algorithm (Intel Pentium D, 3000 MHz)

Number of antibodies, 10^3	20	40	80	100	120	140	160	200
Time, sec	3.11	6.63	14.43	18.19	23.82	27.61	32.97	44.85

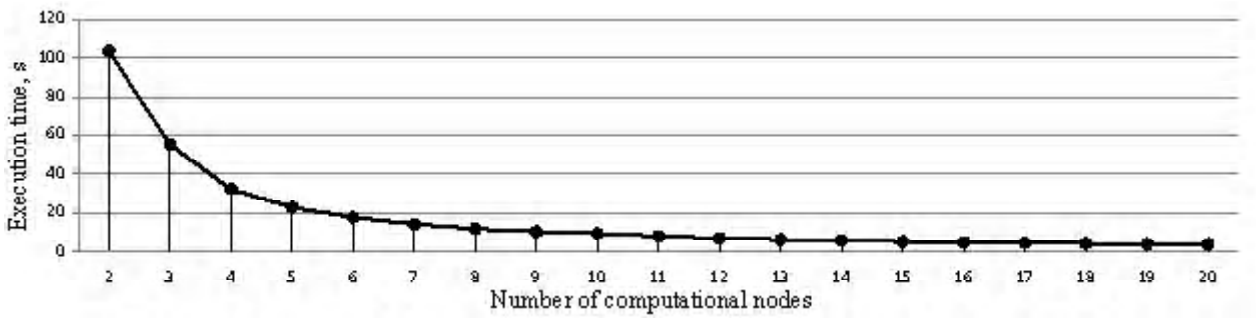

Figure 1 – Dependence of learning time on the number of processors

Table 2 – Acceleration in case of use of TPL on the dual core processor, with various numbers of antibodies in the population

Number of antibodies, 10^3	6	7	8	9	10	11	12	13	14	15
Speedup, %	3.94	4.48	6.0	4.61	14.82	9.49	11.85	7.06	7.04	4.72

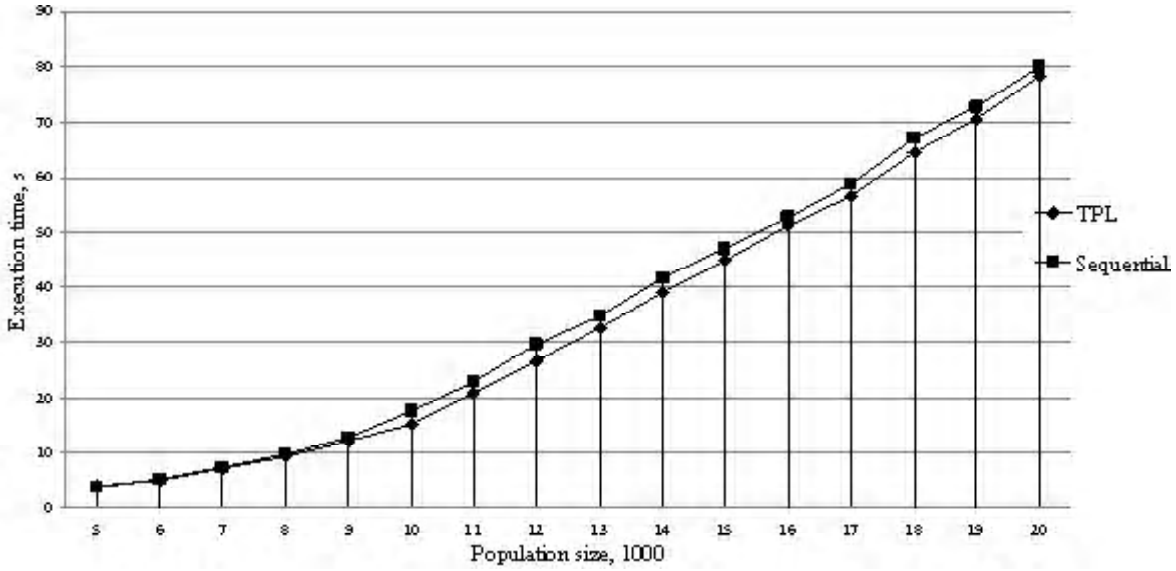

Figure 2 – Dependence of immune algorithm training time on a size of population

forecasting of the distorted time series. However, the growth of a number of missing values requires an increase in the learning sample and the size of antibodies, which leads to the growth of population and requires the use of parallel realization of the immune algorithm, which provides the acceleration close to linear one.

Low efficiency of application of TPL is explained by the fact that in the reviewed modules of the immune algorithm the variables of the most cycles are interdependent that leads to the complication of optimization of the managed code for multi-core processors. Furthermore, in order to get speed gain of parallel program it is necessary that the working time of parallel processes in the areas of code parallelization would exceeds significantly the working hours of generation of parallel flows.

CONCLUSIONS

The paper studied two ways of parallelization of the immune algorithm on the basis of the proposed in [6] hybrid algorithm of short-term time series forecasting. One used a principle of data paralleling on the basis of an island model of genetic algorithm. Were analyzed the considered methods of parallelization on the basis of consideration of communication complexity.

The feature of the first method of parallelization is ability to control the evolution of AIS on computation nodes by the control node, but this leads to a significant communication load. The peculiarity of the second way of parallelization is a minimization of communication overheads within a cluster by means of organization of independent evolutionary process on individual computational nodes during AIS learning. After the processing of training set an interaction between computational nodes and control node takes place only after every integration of algorithm that does not allow in future to use the advantages of a parallel version due to high communication overheads. Since the training time of the system significantly exceeds the receipt time of one forecast, this disadvantage can be neglected.

To implement data exchange between the computational nodes was used MPI.NET technology, to accelerate calculations within the populations of individual nodes was used optimization of managed code for multi-core processor using TPL. Using of the distributed computing allows for acceleration close to linear one, while the code optimization for multi-core architecture allows to obtain the acceleration only about 7 %.

Proposed parallel realizations of the immune algorithms have a high speed of computation of forecasting values and an accuracy of forecasting of different time series comparable with other models.

ACKNOWLEDGEMENTS

This work was performed as part of state budget scientific research work № 273 «Neuro-fuzzy systems for online clustering and classification of data sequences under conditions of distortion by missing and anomalous observations», section № 273-2 «Adaptive techniques and models for data classification and time series prediction in conditions of presence of missing data and outliers, based on artificial immune systems» (state registration number 0113U000361).

REFERENCES

1. Лукашин Ю. П. Адаптивные методы краткосрочного прогнозирования временных рядов / Ю. П. Лукашин. – М. : Финансы и статистика, 2003. – 416 с.
2. Кораблев Н. М. Применение модели клонального отбора, использующей вывод по прецедентам, для прогнозирования временных рядов / Н. М. Кораблев, Г. С. Иващенко // Бионика интеллекта. – № 1(80). – С. 108–111.

3. Dasgupta D. Artificial Immune Systems and Their Applications / D. Dasgupta. – Springer-Verlag, 1999. – 306 p. DOI: 10.1007/978-3-642-59901-9
4. Бидюк П. И. Алгоритм клонального отбора для прогнозирования нестационарных динамических систем / П. И. Бидюк, В. И. Литвиненко, И. В. Баклан, А. А. Фефелов // Искусственный интеллект. – 2004. – № 4. – С. 89–99.
5. Кораблев Н. М. Применение искусственных иммунных сетей для прогнозирования временных рядов / Н. М. Кораблев, Г. С. Иващенко // Системи обробки інформації. – Харьков. – 2012. – № 9(107). – С. 42–45.
6. Кораблев Н. М. Применение модели клонального отбора для прогнозирования временных рядов, имеющих пропущенные значения / Н. М. Кораблев, Г. С. Иващенко // Электротехнические и компьютерные системы. – Одесса. – 2014. – № 13(89). – С. 170–177.
7. Воеводин В. В. Параллельные вычисления. / В. В. Воеводин, Вл.В. Воеводин – СПб.: БХВ-Петербург, 2002. – 475 с.
8. Timothy G. M. Patterns for Parallel Programming / G. M. Timothy, A. S. Beverly, L. M. Berna. – Addison-Wesley, 2004. – 384 p. DOI: 10.1145/1953611.1953620
9. Монахов О. Г. Параллельный генетический алгоритм для оптимизации торговых стратегий / О. Г. Монахов // Сибирский журнал вычислительной математики. – 2008. – Т. 11, № 4. – С. 423–432.
10. Makridakis S. The M-3 Competition: Results, Conclusions and Implications / S. Makridakis, M. Hibon // International of Forecasting. – Amsterdam : Elsevier, 2000. – № 16. – P. 451–476. DOI: 10.1016/S0169-2070(00)00057-1

Article was submitted 03.11.2014.
After revision 05.11.2014.

Кораблев Н. М.[1], Иващенко Г. С.[2]
[1]Д-р техн. наук, профессор, профессор кафедры электронных вычислительных машин, Харьковский национальный университет радиоэлектроники, Украина
[2]Аспирант, Харьковский национальный университет радиоэлектроники, Украина

ПАРАЛЛЕЛЬНЫЙ ИММУННЫЙ АЛГОРИТМ КРАТКОСРОЧНОГО ПРОГНОЗИРОВАНИЯ НА ОСНОВЕ МОДЕЛИ КЛОНАЛЬНОГО ОТБОРА

Рассмотрены способы распараллеливания гибридного иммунного алгоритма краткосрочного прогнозирования временных рядов, построенного на основе модели клонального отбора, использующей вывод по прецедентам и простейшие методы прогнозирования. Проведен анализ двух способов распараллеливания, отличающихся порядком обмена сообщениями между вычислительными узлами. Для реализации предложенного алгоритма используется технология MPI.NET для систем с передачей сообщений. Для оптимизации работы отдельных вычислительных узлов использована библиотека TPL для систем с общей памятью. Представлены результаты экспериментальных исследований, демонстрирующие эффективность предлагаемого подхода.

Ключевые слова: прогнозирование, временной ряд, искусственные иммунные системы, модель клонального отбора, антитело, антиген, аффинность, клонирование, мутация.

Корабльов М. М.[1], Іващенко Г. С.[2]
[1]Д-р техн. наук, професор, професор кафедри електронних обчислювальних машин Харківського національного університету радіоелектроніки, Україна
[2]Аспірант кафедри електронних обчислювальних машин Харківського національного університету радіоелектроніки, Україна

ПАРАЛЕЛЬНИЙ ІМУННИЙ АЛГОРИТМ КОРОТКОСТРОКОВОГО ПРОГНОЗУВАННЯ НА ОСНОВІ МОДЕЛІ КЛОНАЛЬНОГО ВІДБОРУ

Розглянуто способи розпаралелювання гібридного імунного алгоритму короткострокового прогнозування часових рядів, побудованого на основі моделі клонального відбору, яка використовує висновок за прецедентами та найпростіші методи прогнозування. Проведено аналіз двох способів розпаралелювання, що відрізняються порядком обміну повідомленнями між обчислювальними вузлами. Для реалізації запропонованого алгоритму використовується технологія MPI.NET для систем з передачею повідомлень. Для оптимізації роботи окремих обчислювальних вузлів застосована бібліотека TPL для систем із загальною пам'яттю. Представлені результати експериментальних досліджень, що демонструють ефективність запропонованого підходу.

Ключові слова: прогнозування, часовий ряд, штучні імунні системи, модель клонального відбору, антитіло, антиген, афінність, клонування, мутація.

REFERENCES

1. Lukashin Yu. P. Adaptivnye metody kratkosrochnogo prognozirovaniya vremennykh ryadov [Adaptive methods of short-term time series forecasting]. Moskow, Finansy i statistika, 2003, 416 p.
2. Korablev N. M., Ivashchenko G. S. Primenenie modeli klonal'nogo otbora, ispol'zuyushchei vyvod po pretsedentam, dlya prognozirovaniya vremennykh ryadov [Application of the clonal selection model using the case based reasoning for time series forecasting], *Bionika intellekta*. Kharkov, 2013, pp. 108–111.
3. Dasgupta D. Artificial Immune Systems and Their Applications. Springer-Verlag, 1999, 306 p. DOI: 10.1007/978-3-642-59901-9
4. Bidyuk P. I., Litvinenko V. I., Baklan I. V., Fefelov A. A. Algoritm klonal'nogo otbora dlya prognozirovaniya nestatsionarnykh dinamicheskikh sistem [Clonal selection algorithm for the prediction of nonstationary dynamical systems], *Iskusstvennyi intellekt,* 2004, No. 4, pp. 89–99.
5. Korablev N. M., Ivashchenko G. S. Primenenie iskusstvennykh immunnykh setei dlya prognozirovaniya vremennykh ryadov [Application of artificial immune networks for time series forecasting], *Sistemi obrobki informatsii*. Kharkov, 2012, No. 9, pp. 42–45.
6. Korablev N. M., Ivashchenko G. S. Model of clonal selection for forecasting time series with missing data [Primenenie modeli klonal'nogo otbora dlya prognozirovaniya vremennykh ryadov, imeyushchikh propushchennye znacheniya], *Elektrotekhnicheskie i komp'yuternye sistemy*. Odessa, 2014, No.13 (89), pp. 170–177.
7. Voevodin V. V., Voevodin Vl.V. Parallel'nye vychisleniya [Parallel computing] Sankt-Peterburg, BKhV-Peterburg, 2002, 475 p.
8. Timothy G. M., Beverly A. S., Berna L. M. Patterns for Parallel Programming. Addison-Wesley, 2004, 384 p. DOI: 10.1145/1953611.1953620
9. Monakhov O. G. Parallel'nyi geneticheskii algoritm dlya optimizatsii torgovykh strategii [Parallel genetic algorithm for optimization of trading strategies], *Sibirskii zhurnal vychislitel'noi matematiki,* 2008, Vol. 11, No. 4, pp. 423–432.
10. Makridakis S., Hibon M. The M-3 Competition: Results, Conclusions and Implications / S. Makridakis, *International of Forecasting*. Amsterdam, Elsevier, 2000, No. 16, pp. 451–476. DOI: 10.1016/S0169-2070(00)00057-1