ISSN 0146-4116, Automatic Control and Computer Sciences, 2015, Vol. 49, No. 1, pp. 11–20. © Allerton Press, Inc., 2015.
Original Russian Text © A.O. Oliinyk, S.Yu. Skrupsky, S.A. Subbotin, 2015, published in Avtomatika i Vychislitel'naya Tekhnika, 2015, No. 1, pp. 18–30.

Experimental Investigation with Analyzing the Training Method Complexity of Neuro-Fuzzy Networks Based on Parallel Random Search

A. O. Oliinyk, S. Yu. Skrupsky, and S. A. Subbotin
Zaporozhye National Technical University,
Zhukovskogo str. 64, Zaporozhye, 69063 Ukraine
e-mail: subbotin@zntu.edu.ua
Received April 24, 2014

Abstract—The problem of training neuro-fuzzy networks is discussed. The computational complexity of the method for training neuro-fuzzy networks on the basis of parallel random search is analyzed. Theoretical estimations of the speedup and efficiency of the method are found. Software implementing of the method in C++ with using the MPI library and providing the construction of neuro-fuzzy networks in terms of the given observation sets is developed. Experiments for practical tasks are carried out.

Keywords: neuro-fuzzy network, parallel calculations, feature, random search, speedup, efficiency
DOI: 10.3103/S0146411615010071

1. INTRODUCTION

To build models based on the neuro-fuzzy networks in automation of technical and medicine diagnosis, quality control, prediction, and classification, the authors of [1] proposed a parallel method for synthesizing the neuro-fuzzy networks [2–5] by random search [6–9]. In the method, to adjust the parameters of the synthesized models, random optimization [9–12] is used, and the initial set of solutions is formed with considering the learning sample data (the relevance of the feature terms is taken into account by using the density of learning set patterns in the corresponding term and the extent of its influence on the value of the output parameter, which enables approaching the initial point of search to the optimal ones and accelerating the optimization.

Practical application of the method requires the theoretical analysis of its complexity and the experimental investigation of its properties and characteristics at solving practical problems.

The purpose of the present article is the analysis of the computational complexity and experimental investigation of the parametric identification of neuro-fuzzy networks on the basis of parallel random search.

2. FORMULATION OF THE TRAINING PROBLEM OF NEURO-FUZZY NETWORKS

Let the observation (precedent) set S (1) describe the state of the objects and processes under investigation:

$$S = \langle P, T \rangle, \tag{1}$$

where P is the set of observation characteristics (attributes and features), and T is the set of response values (the output parameter).

The sets of P and T are accordingly presented as matrix (2) and vector (3):

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1m} & \cdots & p_{1M} \\ p_{21} & p_{22} & \cdots & p_{2m} & \cdots & p_{2M} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ p_{q1} & p_{q2} & \cdots & p_{qm} & \cdots & p_{qM} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ p_{Q1} & p_{Q2} & \cdots & p_{Qm} & \cdots & p_{QM} \end{pmatrix}, \tag{2}$$

$$T = (t_1 \; t_2 \; \dots \; t_q \; \dots \; t_Q)^{\text{tr}}, \tag{3}$$

where p_{qm} is the value of the m-th feature of the q-th observation ($m = 1,2,\dots,M$, $q = 1,2,\dots,Q$), t_q is the response value of the q-th observation, M is the number of features, Q is the number of observations, and tr is the transposition sign.

The structure of the neuro-fuzzy network is formed by reflecting the sample patterns (1) into the fuzzy productional rules. Then, the training problem of the recognizing model on the basis of the neuro-fuzzy network *NFN* lies in the identification of its parameters (the set of values of the weight coefficients *W* and the set of the parameters of membership function μ) to provide the appropriate value of the given quality criterion *G* of the neuronal model *NFN*. The recognizing error (in the problems with the discrete output *T*) or the mean-square error (if the output parameter *T* can take real values from some range $T \in [t_{\min}; t_{\max}]$) can be used as the target criterion *G* at training the neuro-fuzzy models.

3. ANALYSIS OF THE COMPUTATIONAL COMPLEXITY AND ESTIMATION OF THE PARALLEL SYSTEM EFFICIENCY

To estimate the computational complexity of the method for synthesizing the neuro-fuzzy networks [1], let us define the complexity *O* of each stage, where *O* is the Landau operator "big o" characterizing the number of elementary operations that are necessary to solve some problem. Assume that all the operations are equivalent in terms of the computational complexity.

On initialization, each feature p_m ($m = 1,2,\dots,M$) of the observation set $S = \langle P, T\rangle$ is divided into N_{int} intervals of $\Delta p_{mn} = [p_{mn\min}; p_{mn\max}]$ ($n = 1,2,\dots,N_{\text{int}}$) with the relevance V_{mn} of each term Δp_{mn}. The last one requires defining the entropy $Entr(\Delta p_{mn})$ of each interval Δp_{mn} and the probabilities ρ_q (we will use the denotation system from [1]). Therefore, the computational complexity of the given process can be determined from (4):

$$O(V_{init}) = O(MN_{\text{int}} + MN_{\text{int}}N_{\text{int}T}) = O(MN_{\text{int}}N_{\text{int}T}), \tag{4}$$

where $N_{\text{int}T}$ is the number of decomposition intervals (classes) of the output parameter *T* of set *S*.

Moreover, the formulas are applied for the initialization (calculation) of the values c_{mn}, d_{mn}, and $w_m^{(4,r)}$ ($m = 1,2,\dots,M$, $n = 1,2,\dots,N_{\text{int}}$, $r = 1,2,\dots,N_R$) for each solution $\chi_k^{(0)}$ ($k = 1,2,\dots,N_\chi$) in the set $R^{(0)} = \{\chi_1^{(0)}, \chi_2^{(0)}, \dots, \chi_{N\chi}^{(0)}\}$. The number of elementary operations needed for the solution of the problem can be defined as follows (5):

$$O(\chi_{init}) = O(N_\chi(MN_{\text{int}} + MN_{\text{int}} + MN_R)) = O(N_\chi M(N_{\text{int}} + N_R)). \tag{5}$$

The values of the target functions $G_k^{(0)} = G(\chi_k^{(0)}) = G(NFN_k^{(0)})$ are calculated on N_{pr} parallel processes for each N_χ/N_{pr} solution of $\chi_k^{(0)}$. Finding the estimations $G_k^{(0)}$ is concerned with the need for the transformation $\chi_k^{(0)} \to NFN_k^{(0)}$, which requires processing each observation in the set $S = \langle P, T\rangle$. The computational complexity $O(G)$ of this processes is defined by formula (6):

$$O(G) = O\left(\frac{N_\chi}{N_{pr}} N_{pr}(MN_{\text{int}} + N_R + N_R + MN_R + N_R)Q\right) = O(N_\chi MQ(N_{\text{int}} + N_R)). \tag{6}$$

AT the transformation $R^{(i)} \to \{R^{(i,1)}, R^{(i,2)}, \dots, R^{(i,N_{pr})}\}$, the intervals of the possible values of each N_g of adjusted parameters c_{mn}, d_{mn}, and $w_m^{(4,r)}$ are divided on N_{pr} sections for each N_χ of $\chi_k^{(0)}$ solutions. The number of operations required for the transformation $R^{(i)} \to \{R^{(i,1)}, R^{(i,2)}, \dots, R^{(i,N_{pr})}\}$ is estimated as follows (7):

$$O(R_{init}) = O(N_g N_\chi). \tag{7}$$

Clearly, the most complicated problem is the calculation of the values of the target function (the $O(G)$ estimation) being proportional to the product of the values $N_\chi MQ(N_{\text{int}} + N_R)$, which causes the advisability of paralleling this process.

According to (4)–(7), as well as $N_{\text{int}T} \ll Q$ (the number of decomposition intervals of the output parameter T in the set $S = \langle P, T \rangle$ is significantly smaller that the general number of observations Q in this set) and $N_g \ll MQ$ (the number of the adjusted parameters of the model is significantly smaller than the general number of the elements of S), the general complexity O_{init} of the initialization can be estimated by (8):

$$\begin{aligned} O_{init} &= O(V_{init}) + O(\chi_{init}) + O(G) + O(R_{init}) \\ &= O(MN_{\text{int}}N_{\text{int}T} + N_\chi M(N_{\text{int}} + N_R) + N_\chi MQ(N_{\text{int}} + N_R) + N_g N_\chi) = O(N_\chi MQ(N_{\text{int}} + N_R)). \end{aligned} \tag{8}$$

The random search $RS(R^{(i,j)})$ is needed for processing the sets consisting of the N_χ / N_{pr} solutions. (The selection of the solutions, their crossing, and their mutation, as well as the calculation of the target functions, should be implemented on each N_{pr}). Estimating the computational complexity of each process can be carried out by (9), (10), (11), and (6):

$$O\left(RS^{(i,j)}_{\text{select}}\right) = O\left(\frac{N_\chi}{N_{pr}} N_{pr}\right) = O(N_\chi), \tag{9}$$

$$O\left(RS^{(i,j)}_{\text{cross}}\right) = O\left(\beta \frac{N_\chi}{N_{pr}} N_g N_{pr}\right) = O(\beta N_\chi N_g), \tag{10}$$

$$O\left(RS^{(i,j)}_{\text{mutation}}\right) = O\left(\gamma\vartheta \frac{N_\chi}{N_{pr}} N_g N_{pr}\right) = O(\gamma\vartheta N_\chi N_g), \tag{11}$$

$\beta + \gamma < 1$, $\vartheta < 1$.

Therefore, the estimation of the computational complexity of the process $RS(R^{(i,j)})$ at N_{it} iterations can be defined as follows (12):

$$\begin{aligned} O\left(RS^{(i,j)}\right) &= O(N_{it}(N_\chi + \beta N_\chi N_g + \gamma\vartheta N_\chi N_g + N_\chi MQ(N_{\text{int}} + N_R))) \\ &= O(N_{it} N_\chi MQ(N_{\text{int}} + N_R)). \end{aligned} \tag{12}$$

The number of operations required for random search $RS(R^{(i)})$ over the generalized population $R^{(i)}$ with taking into account the fact that the values of the target functions are calculated on parallel processes is defined in a similar manner (13):

$$\begin{aligned} O\left(RS^{(i)}\right) &= O(N_{it}(N_\chi + \beta N_\chi N_g + \gamma\vartheta N_\chi N_g + N_\chi MQ(N_{\text{int}} + N_R))) \\ &= O(N_{it} N_\chi MQ(N_{\text{int}} + N_R)). \end{aligned} \tag{13}$$

Since the processes $RS(R^{(i,j)})$ and $RS(R^{(i)})$ are alternately implemented (assume that these processes can repeat not more than by N_{it} iterations) up to the satisfaction of the stopping criteria, the general estimation of the computational complexity is defined by formula (14):

$$O = O_{init} + N_{it}\left(O\left(RS^{(i,j)}\right) + O\left(RS^{(i)}\right)\right). \tag{14}$$

Let us estimate the maximum possible speedup $Speedup_{pr}$ and the efficiency of the parallel system $Efficiency_{pr}$ upon using the proposed method of the parametric synthesis of the neuro-fuzzy models on theN_{pr} processes of the parallel system.

Let α_{serial} be the calculation portion on the sequential stages of the method. The portion of them for the method α_{serial} is determined by (15):

$$\alpha_{\text{serial}} = \frac{N_{\text{serial}}}{N_{\text{serial}} + N_{\text{paral}}} = \frac{1}{1 + \dfrac{N_{\text{paral}}}{N_{\text{serial}}}}, \tag{15}$$

where N_{serial} is the number of sequential operations, and N_{paral} is the number of parallel operations.

We will use the previously defined estimations of the computational complexity of different operations of the proposed method as the ones of N_{serial} and N_{paral} by means of (4)–(14).

This method implies the sequential implementation of some operations on initialization (the calculation complexity of such operations according to the above mentioned formula is estimated as $O(V_{init})$, $O(\chi_{init})$, $O(R_{init})$) and the ones concerned with generating a new set of solutions (selection, crossing, and mutation) at the random search $RS(R^{(i)})$ (the computational complexity of such operations as $N_{it}(O(RS^{(i)}_{\text{select}}) + O(RS^{(i)}_{\text{cross}}) + O(RS^{(i)}_{\text{mutation}}))$. Therefore, the order of the estimation of N_{serial} is defined as follows (16):

$$N_{\text{serial}} \sim O(V_{init}) + O(\chi_{init}) + O(R_{init}) + N_{it}\left(O\left(RS^{(i)}_{\text{select}}\right) + O\left(RS^{(i)}_{\text{cross}}\right) + O\left(RS^{(i)}_{\text{mutation}}\right)\right). \tag{16}$$

On initialization, the operations concerned with calculating the values of the target function ($O(G) = O(N_\chi MQ(N_{\text{int}} + N_R))$) and the processes of the random search $RS(R^{(i,j)})$ with the estimations of the computational complexity

$$O\left(RS^{(i,j)}\right) = O\left(N_{it}\left(N_\chi + \beta N_\chi N_g + \gamma\vartheta N_\chi N_g + N_\chi MQ(N_{\text{int}} + N_R)\right)\right),$$

as well as the calculation of the target function G at $RS(R^{(i)})$ (the computational complexity is $O(RS^{(i)}_G) = O(N_{it}N_\chi MQ(N_{\text{int}} + N_R)))$, are implemented. Consequently, the order of the value N_{paral} is found as follows (17):

$$\begin{gathered} N_{\text{paral}} \sim O(G) + N_{it}\left(O\left(RS^{(i,j)}\right) + O\left(RS^{(i)}_G\right)\right) = \left(N_\chi MQ(N_{\text{int}} + N_R)\right) \\ + N_{it}O\left(N_{it}\left(N_\chi + \beta N_\chi N_g + \gamma\vartheta N_\chi N_g + N_\chi MQ(N_{\text{int}} + N_R)\right) + N_{it}\left(N_\chi MQ(N_{\text{int}} + N_R)\right)\right). \end{gathered} \tag{17}$$

As mentioned above, the estimations $O(V_{init})$, $O(\chi_{init})$, $O(R_{init})$, and, therefore, their sum $O(V_{init}) + O(\chi_{init}) + O(R_{init})$ are significantly lower than $O(G) = O(N_\chi MQ(N_{\text{int}} + N_R))$. Substituting (16) and (17) in (15), we obtain the estimation (18) of the calculation portion α_{serial} in the sequential stages of the method. The sum $O(V_{init}) + O(\chi_{init}) + O(R_{init})$ in (16) is neglected since it is as least by N_{it}^2 times lower than (17). Here, we will take into account $O(G)$ because it is considerably smaller than $N_{it}(O(RS^{(i,j)}) + O(RS^{(i)}_G))$:

$$\begin{aligned} \alpha_{\text{serial}} &\sim \frac{1}{1 + \dfrac{N_{it}\left(O\left(RS^{(i,j)}\right) + O\left(RS^{(i)}_G\right)\right)}{N_{it}\left(O\left(RS^{(i)}_{\text{select}}\right) + O\left(RS^{(i)}_{\text{cross}}\right) + O\left(RS^{(i)}_{\text{mutation}}\right)\right)}} \\ &= \frac{1}{1 + N_{it}\dfrac{O\left(1 + \beta N_g + \gamma\vartheta N_g + 2MQ(N_{\text{int}} + N_R)\right)}{O\left(1 + \beta N_g + \gamma\vartheta N_g\right)}} = \frac{1}{1 + N_{it}\left(1 + 2\dfrac{O\left(MQ(N_{\text{int}} + N_R)\right)}{O\left(1 + \beta N_g + \gamma\vartheta N_g\right)}\right)}. \end{aligned} \tag{18}$$

As noted above, the number of adjusted parameters N_g of the model is significantly smaller than the general number of elements of the set $S = \langle P, T\rangle$ ($N_g \ll MQ$) since the synthesized model should possess the generalization properties. Let us consider that the proposed parallel method of the parametric synthesis of the neuro-fuzzy networks is used to adjust the $N_g \sim 10^2$ parameters of the neural model by a sampling containing $Q \sim 10^2$ observations, which are described by $M \sim 10^1$ features. Here, the number of rules is $N_R \sim 10^1$, the number of the decomposition intervals of the features is $N_{\text{int}} \sim 10^1$, and the maximal admissible number of iterations of the random search is $N_{it} \sim 10^1$. Putting the orders of the corresponding values into (18), we found the order of the portion of calculations α_{serial}, falling in the sequential stages of the method (19):

$$\alpha_{\text{serial}} \sim \frac{1}{1 + N_{it}\left(1 + 2\dfrac{O\left(MQ(N_{\text{int}} + N_R)\right)}{O\left(1 + \beta N_g + \gamma\vartheta N_g\right)}\right)} \approx 10^{-3}. \tag{19}$$

It should be noted that, since the orders of N_{int}, N_R, and N_{it} are generally higher than those used at estimating α_{serial}, the order of α_{serial} can be significantly smaller than the one in (19) in practice.

As seen from (19), the portion of the sequential operations $\alpha_{\text{serial}} \approx 10^{-3}$ is negligibly small. The portion of the parallel operations α_{paral} is defined as follows (20):

$$\alpha_{\text{paral}} = 1 - \alpha_{\text{serial}} \approx 1 - 10^{-3} = 0.999. \tag{20}$$

The theoretical speedup $Speedup_{pr}$ of the parametric synthesis of the neuro-fuzzy models, which can be obtained with applying the method on the N_{pr} processes of the parallel system, is determined according to Amdahl's law [13] (21):

$$Speedup_{pr} = \frac{1}{\alpha_{\text{serial}} + \dfrac{1-\alpha_{\text{serial}}}{N_{pr}}} \approx \frac{1}{10^{-3} + \dfrac{1-10^{-3}}{N_{pr}}} = \frac{N_{pr}}{10^{-3}N_{pr} + 0.999}. \tag{21}$$

At $N_{pr} = 16$ and $N_{pr} = 4$, the value of the theoretical speedups of the parallel system are $Speedup_4 = 3.988$ and $Speedup_{16} = 15.76$, respectively. Such values of $Speedup_{pr}$ show the theoretical ability of the developed method to demonstrate significant speedup upon its using in the parallel system.

The efficiency $Efficiency_{pr}$ of the parallel system is defined by using the proposed method (22):

$$Efficiency_{pr} = \frac{Speedup_{pr}}{N_{pr}} = \frac{1}{N_{pr}}\frac{N_{pr}}{10^{-3}N_{pr} + 0.999} = \frac{1}{10^{-3}N_{pr} + 0.999}. \tag{22}$$

The efficiencies are $Efficiency_4 = 0.997$ and $Efficiency_{16} = 0.985$ at $N_{pr} = 4$ and $N_{pr} = 16$ respectively. Clearly, the value of $Efficiency_{pr}$ at different N_{pr} is close to unity, which proves the high efficiency of the parallel method of the parametric synthesis of the neuro-fuzzy models.

However, since Amdahl's law does not take into account the expenses concerned with the communication between the processes (information forwarding), the real speedup will be smaller than the estimated one. To define the corresponding characteristics, it is reasonable to implement the experimental investigation of the proposed method.

4. EXPERIMENTAL PROCEDURE AND THE RESULTS

To carry out the experimental investigation, the cluster of the Pukhov Institute for Modelling in Energy Engineering of the National Academy of Sciences of Ukraine was used. There were 16 logical nodes (each of them implements one process) of the following configuration: the processor Intel Xeon 5405, the RAM 4×2 GB DDR-2 on each computational node, and the communication medium InfiniBand 20Gb/s. Intermediate software, namely, the planner Torque and the package OMPI, was installed in the cluster nodes.

For the experimental estimation of the proposed method, software based on C++ with using MPI was developed [14, 15]. The time was measured by the library function MPI_Wtime. The experiments were carried out five times on one, four, six, ten, twelve, fourteen, and sixteen processes.

The method and software was applied to solve the problem of diagnosing aircraft gas turbine engine blades [16] for the purpose of increasing the reliability and life time of the aircraft engines.

The learning sample $S = \langle P, T\rangle$ contained information about 354 observations (blades), each of which was a set of values of the power spectra of free damped oscillations of the impact. The output parameter T of each blade P_q involved the blade condition data (the zero value corresponded to the defective blades and the unit one to the serviceable blades). The state of each blade P_q was defined by implementing the impact excitation and recording the signals, which were subjected to spectral analysis, resulting in the generation of 2610 signal characteristics, namely, the values of the mth feature of p_{qm} of the qth blade.

The task was in the construction of the diagnostic model allowing one to recognize the defective blades of the aircraft engines by the values of the power spectra of the free damped oscillations after the impact excitation. Since it was required to create a model providing a zero error level, there were no early outputs from the processes of the random search; the stochastic optimization terminated at the achievement of the maximum permissible quantity of iterations (this enabled studying the different characteristics of the proposed method in more detail). The developed neuro-fuzzy model parametric identification parallel method (NFMPIPM) was compared with the Island Genetic Algorithm (IGA) [10–12]).

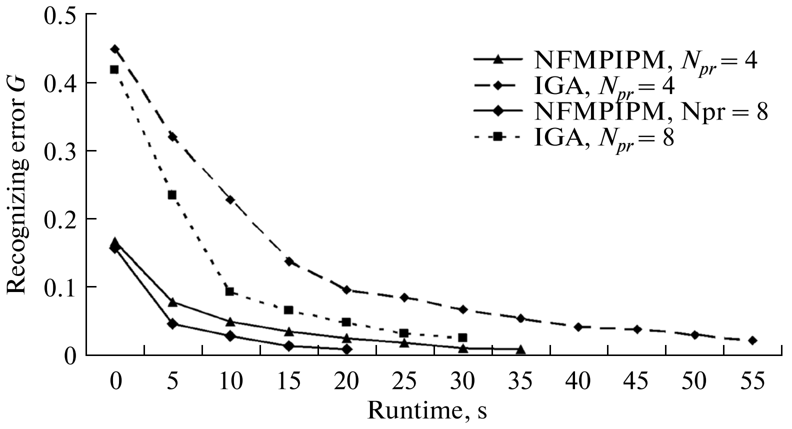

Fig. 1. Plot of the relationship between the recognizing error of the neuro-fuzzy model and the operation time of the methods.

Figure 1 shows the recognizing error G of the synthesized neuro-fuzzy model *NFN* in dependence on the implementation time of NFMPIPM and IGA at different numbers of processes N_{pr} of the parallel system.

As seen from Fig. 1, the proposed method allows one to achieve admissible results quickly. For example, the developed method provided constructing a neuro-fuzzy network with the error level $G = 0.0085$ at $N_{pr} = 4$ for 36.63 s. When the island model of the evolutionary search is used, a neuro-fuzzy model characterized by the error $G = 0.023$ was synthesized for 53.16 s. All the methods allowed one to build a model with a zero error level; therefore, the search process terminated at the stopping criterion to achieve the maximum permissible number of iterations. As opposed to the IGA method, the shortening of the parallel random search time in the proposed method is causes by the following factors:

—Using the a prior information about the learning sample by taking into account the relevance of the terms of the features on the basis of the density of the patterns of the learning set in the corresponding terms and the extent of their influence on the output parameter. Figure 1 shows that both methods (NFMPIPM and IGA) on the initial search iterations provided constructing neuro-fuzzy models characterized by approximately identical values of the recognizing error G regardless of the number of processes N_{pr} of the parallel system. The error G of the synthesized neuro-fuzzy models on the initial iterations was significantly smaller than the error of the IGA models. This is due to using the a prior information of the learning sample on initialization, which enables approaching the initial search points to the optimal ones and accelerating the optimization. Applying the island model of the evolutionary search IGA, the initial population is formed in a random way, which affects the values of the recognizing error of the neuro-fuzzy networks and, therefore, requires larger computational expenses in searching for the parameters of the synthesized models.

—The presence of the supporting mechanisms of the diversity in the population that provided the search. This resulted in achieving a more appropriate error of the synthesized neuro-fuzzy model for a smaller time in comparison to IGA.

—The application of the modified operators for generating new solutions by crossing and mutation, wherein the location features of the adjusted parameters, their type (the membership function parameters of neurons and the weight coefficients of the neuroelement), and the suitability of the parent solutions are taken into account. Moreover, overrunning is allowed, which results in expanding the search area and saving time and computational resources at cycling in the local extremum regions.

Table 1 gives the efficiency features of the cluster implementing the parametric synthesis of the neuro-fuzzy networks with using the proposed method.

Figure 2 portrays the averaged time burden for constructing the neuro-fuzzy models by different parallel methods.

Figure 2 shows that NFMPIPM enables implementing the parallel random search faster than IGA since the application of the modified operators of generating new solutions decreases the general number of operations on processes.

Figures 3 and 4 gives the speedup and efficiency plots for the parallel system in the proposed and IGA methods.

Table 1. Efficiency features of the cluster implementing the present method

Number of processes, N_{pr}	Runtime, s		Speedup, $Speedup_{pr}$		Efficiency, $Efficiency_{pr}$	
	NFMPIPM	IGA	NFMPIPM	IGA	NFMPIPM	IGA
1	140.87	208.12	1.00	1.00	1.00	1.00
2	71.84	105.93	1.96	1.96	0.98	0.98
4	36.63	53.96	3.85	3.84	0.96	0.95
6	24.89	36.46	5.66	5.63	0.94	0.93
8	19.19	27.89	7.34	7.32	0.92	0.91
10	15.64	22.89	9.01	8.80	0.90	0.88
12	13.38	19.48	10.53	10.27	0.88	0.86
14	11.67	16.98	12.07	11.65	0.86	0.83
16	10.48	15.30	13.45	12.80	0.84	0.8

As seen from these plots, the speedup almost linearly grows in proportion to the number of processes. The slow decrease in the system's efficiency with increasing the number of processes proves the good scalability of the proposed method on parallel architecture MPP (massive parallel systems) [15]. Compared to IGA, the present method showed better scalability. This is due to the following reasons:

—less time for the process synchronization;

—faster preparation of the data for forwarding.

The main reason for the speedup decrease is the forwardings. To estimate their portion between the processes, the calls of the MPI_Wtime functions were implemented directly before and after each forwarding. During the analysis of the method, it was found that the migration frequency affects the number of forwardings: the more frequent the migrations are, the larger the number of forwardings between the processes are required. The percentage of the migrating specimens also influences the forwarded data volume.

To estimate the effect of the forwardings on the computational process, experiments on varying the migration frequency and the specimen percent and finding the forwarding time with respect to the number of processes were carried out.

Tables 2 and 3 give the results of estimating the time and forwarding portion at different migration frequencies L_{migr}, as well as the percentage of migrating specimens L_{Hmigr}.

Clearly, the growth of the migration frequency increases the time and portion of the forwardings in relation to the target calculation on the cluster to a greater extent than the growth of the migrating specimens. This is owing the longer time of the initialization of the forwardings in the parallel medium in comparison with the forwardings themselves.

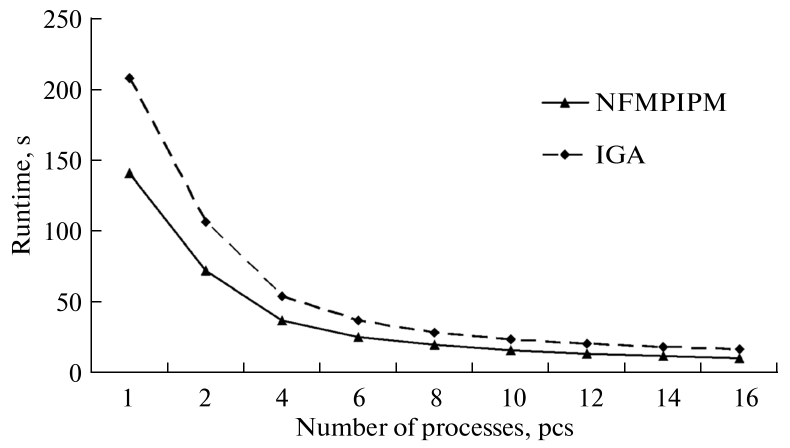

Fig. 2. Averaged time expenses of implementing the methods in the parallel system.

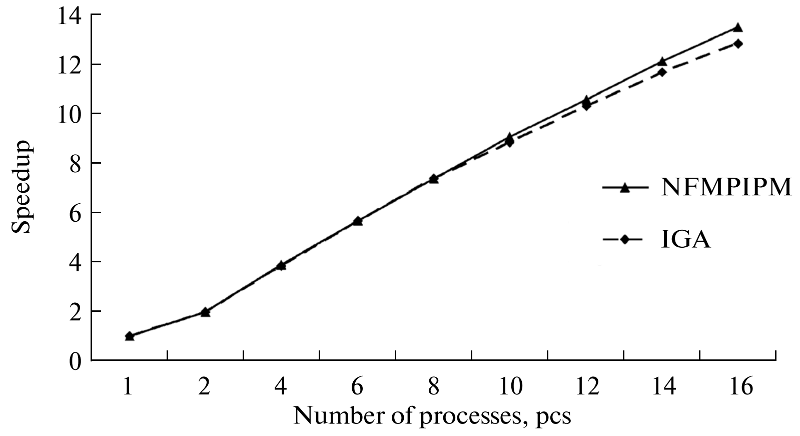

Fig. 3. Plot of the parametrical synthesis speedup of neuro-fuzzy networks by the NFMPIPM and IGA methods.

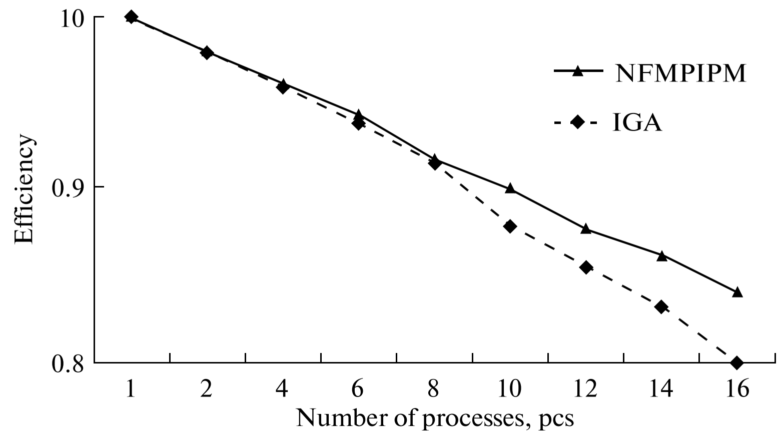

Fig. 4. Plot of the efficiency of the parallel system implementing the parametrical synthesis of the neuro-fuzzy networks with using the proposed and IGA methods.

Figure 5 depicts the relationship between the forwarding portion and the number of processes at $L_{Hmigr} = 2\%$.

Table 2. Forwarding time between the processes at different migration frequencies L_{migr} and portions of the migrating specimens L_{Hmigr}

Number of processes, N_{pr}	Forwarding time, s							
	$L_{migr} = 1/25$		$L_{migr} = 1/18$		$L_{migr} = 1/25$		$L_{migr} = 1/18$	
	$L_{Hmigr} = 2\%$		$L_{Hmigr} = 2\%$		$L_{Hmigr} = 5\%$		$L_{Hmigr} = 5\%$	
	NFMPIPM	IGA	NFMPIPM	IGA	NFMPIPM	IGA	NFMPIPM	IGA
1	0	0	0.00	0.00	0.00	0.00	0.00	0.00
2	1.44	1.44	1.73	1.73	1.61	1.61	1.91	1.91
4	1.47	1.47	1.80	1.80	1.68	1.68	2.02	2.02
6	1.49	1.49	1.91	1.91	1.76	1.76	2.19	2.19
8	1.73	1.73	2.26	2.26	2.11	2.11	2.66	2.66
10	1.72	1.72	2.30	2.30	2.14	2.14	2.75	2.75
12	1.87	1.87	2.57	2.57	2.39	2.39	3.12	3.12
14	1.87	1.87	2.64	2.64	2.44	2.44	3.26	3.26
16	1.99	1.99	2.94	2.94	2.67	2.67	3.70	3.70

Table 3. Forwarding portion between the processes at different migration frequencies L_{migr} and portions of the migrating specimens L_{Hmigr}

Number of processes, N_{pr}	Forwarding portion							
	$L_{migr} = 1/25$		$L_{migr} = 1/18$		$L_{migr} = 1/25$		$L_{migr} = 1/18$	
	$L_{Hmigr} = 2\%$		$L_{Hmigr} = 2\%$		$L_{Hmigr} = 5\%$		$L_{Hmigr} = 5\%$	
	NFMPIPM	IGA	NFMPIPM	IGA	NFMPIPM	IGA	NFMPIPM	IGA
1	0	0	0	0	0	0	0	0
2	0.02	0.018	0.024	0.020	0.022	0.02	0.026	0.023
4	0.04	0.037	0.049	0.041	0.046	0.041	0.054	0.049
6	0.06	0.051	0.076	0.059	0.07	0.057	0.086	0.069
8	0.09	0.072	0.115	0.082	0.108	0.082	0.133	0.1
10	0.11	0.100	0.143	0.11	0.134	0.117	0.167	0.143
12	0.14	0.123	0.185	0.124	0.174	0.146	0.218	0.18
14	0.16	0.142	0.216	0.190	0.202	0.172	0.258	0.213
16	0.19	0.176	0.264	0.183	0.244	0.216	0.319	0.271

The forwarding portion grows faster with increasing the number of processes. When it exceeds the portion of the target values, the efficiency of such a system will become low.

The proposed NFMPIPM and the IGA method showed the same time for forwarding the data between the processes, and the portion of such forwardings from the general computational process in IGA is smaller.

As seen from Tables 2 and 3, the growth of the migration frequency and the percentage of the migrating specimens significantly increase the portion of forwardings, which decreases the efficiency of the parallel system in relation to the computational time. Therefore, it is advisable to select low values of the migration frequency and specimen partition parameters in practice to reduce the computational time for the cluster.

Thus, the experiments implemented on the cluster prove the appropriate values of the efficiency features of the present parallel method of the parametric synthesis of neuro-fuzzy models.

5. CONCLUSIONS

The authors of the present article solved the currently central problem of analyzing the complexity and investigating the method for the synthesis of neuro-fuzzy models by the given sets of observations in terms of parallel evolutionary search [1].

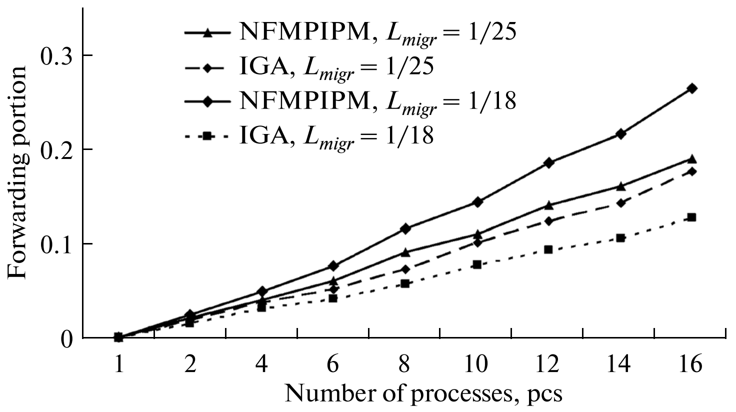

Fig. 5. Relationship between the forwarding portion between the processes for the NFMPIPM and IGA methods at different migration frequencies L_{migr}.

The scientific novelty of the results is in the theoretical estimations of the computational complexity of the method proving the high efficiency of the parallel approach and random search to construct the neuromodels based on the sets of precedents.

The practical value of the results is in the software implementing the proposed method in C++ with applying the MPI library and providing the construction of the neuro-fuzzy networks on the basis of the given observation sets, which was developed and then tested upon solving the diagnosis problem of the aircraft gas turbine engine blades with using the results of signal processing after impact excitation.

The experiment results prove the acceptability of the efficiency parameters of the parallel method of synthesizing the neuro-fuzzy networks.

ACKNOWLEDGMENTS

This work was supported by the state budget research theme of Zaporozhye National Technical University: Intellectual Information Technologies of Automation, Modeling Control, and Diagnosis of Industrial Processes and Systems (state registration no. 0112U005350).

REFERENCES

1. Oliinyk, A.O., Skrupsky, S.Yu., and Subbotin, S.A., Using Parallel Random Search to Train Fuzzy Neural Networks, *Autom. Cont. Compt. Sci.,* 2014, vol. 47, no. 6.
2. Cirstea, M.N., Dinu, A., Khor, J.G., and McCormick, M., *Neural and Fuzzy Logic Control of Drives and Power Systems,* Oxford: Newnes, 2002.
3. Nauck, D., Klawonn, F., and Kruse, R., *Foundations of Neuro-Fuzzy Systems,* Chichester: Wiley, 1997.
4. Engelbrecht, A., *Computational Intelligence: An Introduction,* Sidney: Wiley, 2007.
5. Rutkowski, L., *Flexible Neuro-Fuzzy Systems: Structures, Learning and Performance Evaluation,* Boston: Kluwer, 2004.
6. Baragona, R., Battaglia, F., and Poli, I., *Evolutionary Statistical Procedures: An Evolutionary Computation Approach to Statistical Procedures Designs and Applications,* Berlin: Springer-Verlag, 2011.
7. Abraham, A., Grosan, C., and Pedrycz, W., *Engineering Evolutionary Intelligent Systems,* Berlin: Springer-Verlag, 2008.
8. Shin, Y.C. and Xu,C., *Intelligent Systems: Modeling, Optimization, and Control,* Boca Raton: CRC Press, 2009.
9. Clarke, B., Fokoue, E., and Zhang, H.H., *Principles and Theory for Data Mining and Machine Learning,* New York: Springer-Verlag, 2009.
10. Sumathi, S. and Paneerselvam, S., *Computational Intelligence Paradigms: Theory and Applications using MATLAB,* Boca Raton: CRC Press, 2010.
11. Yu, X. and Gen, M., *Introduction to Evolutionary Algorithms (Decision Engineering),* London: Springer-Verlag, 2010.
12. Cantu-Paz, E., *Efficient and Accurate Parallel Genetic Algorithms,* Massachusetts: Kluwer Academic, 2001.
13. Gebali, F., *Algorithms and Parallel Computing,* New Jersey: Wiley, 2011.
14. Quinn, M.J., *Parallel Programming in C with MPI and Open MP,* New York: McGraw-Hill, 2004.
15. Roosta, S.H., *Parallel Processing and Parallel Algorithms: Theory and Computation,* New York: Springer-Verlag, 2000.
16. Boguslaev, A.V., Oliinyk, O.O., Oliinyk, A.O., Pavlenko, D.V., and Subbotin, S.A., *Progressivnye tekhnologii modelirovaniya, optimizatsii i intellektual'noi avtomatizatsii etapov zhiznennogo tsikla aviatsionnykh dvigatelei,* (Progressive Technologies of Simulation, Optimization and Intellectual Automation of Aviation Motor Life Cycle Stages), Zaporozh'e: OAO "Motor Sich", 2009.

Translated by A. Evseeva