

**ABOUT THE EFFECTIVENESS OF EVOLUTIONARY ALGORITHMS
FOR MULTICRITERIAL DESIGN OF ARTIFICIAL NEURAL NETWORKS**

A. A. Koromyslova*, M. E. Semenina

Siberian State Aerospace University named after academician M. F. Reshetnev
31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660014, Russian Federation

*E-mail: akoromyslova@mail.ru

Artificial neural networks can be widely used in various fields: economics, medicine, space grown, etc. However, using neural networks to solve a particular problem arises the problem of choosing an effective structure of neural networks. Solving these problems is an important step in the application of neural network technology to practical problems, since these stages directly affects the quality (value) of the resulting neural network model. However, this takes more time and material resources, which leads to the need to automate the process. For this purpose the use of multicriteria evolutionary algorithms, such as SPEA, SPEA2 and NSGAI is offered as they can solve two problems at once. Firstly, they can generate a neural network, thus saving computational resources. And secondly, they can solve tasks quite efficiently.

Modified evolutionary algorithms that produce selection of the most informative features, do not improve performance of algorithms that use all the inputs on the problems of small dimension, but significantly improve the accuracy, increasing dimension.

The modified algorithms together with automatic design structure of artificial neural networks determine the most informative features, and include as inputs only weakly correlated with each other variables of the original problem.

Keywords: artificial neural networks design, evolutionary algorithms, multicriteria optimization, most informative features, classification.

Вестник СибГАУ
Т. 16, № 1. С. 79–85

**ОБ ЭФФЕКТИВНОСТИ ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ
МНОГОКРИТЕРИАЛЬНОГО ПРОЕКТИРОВАНИЯ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ**

А. А. Коромыслова*, М. Е. Семенкина

Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева
Российская Федерация, 660014, г. Красноярск, просп. им. газ. «Красноярский рабочий», 31

*E-mail: akoromyslova@mail.ru

Искусственные нейронные сети могут быть широко использованы в различных областях: экономике, медицине, космической отрасли и т. п. Однако при использовании нейронных сетей для решения конкретной задачи возникает проблема выбора эффективной структуры нейронной сети. Решение этих проблем является важным этапом применения нейросетевых технологий для практических задач, так как от этих этапов напрямую зависит качество (адекватность) полученной нейросетевой модели. Однако это требует больших затрат временных и материальных ресурсов, что приводит к необходимости автоматизировать процесс. Для этого предлагается использовать многокритериальные эволюционные алгоритмы, такие как SPEA, SPEA2 и NSGAI, так как они могут решать сразу две проблемы: во-первых, генерировать небольшие нейронные сети, тем самым экономя вычислительные ресурсы, а во-вторых, решать поставленные задачи достаточно качественно.

Модифицированные эволюционные алгоритмы, которые производят отбор наиболее информативных признаков, не улучшают работу алгоритмов, использующих все входы на задачах малой размерности, но значительно повышают точность при росте размерности.

Модифицированные алгоритмы одновременно с автоматическим проектированием структуры искусственных нейронных сетей определяют наиболее информативные признаки, причем включают в качестве входов только слабо коррелированные друг с другом переменные исходной задачи.

Ключевые слова: искусственные нейронные сети, эволюционные алгоритмы, многокритериальная оптимизация, наиболее информативные признаки, классификация.

Introduction. The intensity of the use of intelligent information technologies (IIT) [1] is increasing in all areas of human activity. This is not only due to increasing computing power, which can be used to solve challenging real-world problems, but also due to the ability of systems, based on the use of IIT, to deal effectively with a wide range of tasks: pattern recognition, classification, function approximation, prediction and control [2].

Usually, the implementation of IIT is a time-consuming and complex process. If a researcher decides to use artificial neural networks (ANN) [3] to solve a real world problem, he/she will face the problem of choosing the ANN structure. In contrast to the tuning of weighting coefficients, this issue is not so widely discussed in scientific papers.

In real world problems the dimension can be high, so there is a need to implement the pre-processing of data to reduce the amount of computation effort. An automated choice of the most informative features allows the researcher to keep the performance at an acceptable level using fewer resources.

Since in the design of artificial neural networks it is often difficult to find a compromise between accuracy and the simplicity of the solution obtained by the networks, researchers are encouraged to use evolutionary algorithms of multicriteria optimization, such as Non-dominated Sorting Genetic Algorithm II (NSGAI) [4], Strength Pareto Evolutionary Algorithm (SPEA) [5], Strength Pareto Evolutionary Algorithm 2 (SPEA2) [5]. The effectiveness of these techniques is studied in this paper.

1. Evolutionary Algorithms for Optimization

Genetic algorithm for unconditional one-criterion optimization. Genetic algorithms (GA) belong to the class of adaptive stochastic optimization algorithms [6]. For solving optimization problems a GA allocates a fixed amount of resources determined by the number of individuals in a population and the number of generations. The evaluation of GA effectiveness was fulfilled for different combinations of genetic operators. The identifying of the best settings was carried out by comparing indicators such as the reliability of the algorithm and the average number of generations (iterations), for which a solution was found with a specified accuracy. The reliability is the proportion of GA successful runs. The algorithm with the highest reliability was considered as the best one. Combinations of settings with the same reliability were compared with the second indicator: the smaller the average number of iterations is, the more effective is the algorithm. The required accuracy is equal to 0.01. The results obtained were averaged over 100 runs. The reliability values averaged over all tasks as well as their variations are given in tab. 1. The last column shows the average generation numbers when solutions were found and the scatter throughout all tasks. The effectiveness of the algorithm was verified on an international set of test problems [7].

The results of GA for test problems of unconditional optimization

	Reliability	Generations
The best settings GA	0.937 [0.471; 1]	21.7 [15; 32]
Medium setting GA	0.722 [0.214; 0.961]	14.2 [12; 45]
The worst setting GA	0.55 [0.147; 0.92]	52.9 [14; 69]

Table 1

Analysis of results showed that usually proportional selection, two-point crossover and average mutation are the best settings. Sometimes the best algorithm includes the one-point crossover. The algorithm with tournament selection (tournament size is equal to 3), one-point crossover and low mutation is the worst one.

Evolutionary algorithms for unconditional multicriteria optimization. In its most general form, solving the problem of multicriteria optimization requires the finding of an optimal set of K criteria. Non-dominated points in the domain are called the Pareto set, and their image in the space of criteria is the Pareto front. Usually in multicriteria optimization problems it is sufficient to choose a solution from the Pareto set; these points cannot be preferred to one another but are better than any others. So after the formation of the Pareto set representative approximation the task is considered to be solved [8].

There are many variants of evolutionary algorithms, which can be used for solving multicriteria optimization problems. This paper discusses the Non-dominated sorting Genetic Algorithm II (NSGAI) [4], Strength Pareto Evolutionary Algorithm (SPEA) and Strength Pareto Evolutionary Algorithm 2 (SPEA2) [5].

The efficiency of the algorithms was investigated on the international set of test multicriteria optimization problems [9]. Summary results are presented in tab. 2, which shows the evaluation of the effectiveness of the multicriteria optimization algorithms. Algorithm settings are considered to be the “best” if the solutions found received more points in the Pareto front, as well as if the variation in the spaces of alternatives and criteria is maximized. Tab. 2 also shows estimations of the algorithms with settings that were the worst according to the above criteria and the result averaged over all settings.

The evaluation of the effectiveness of algorithms was performed using three metrics: percentage points in the external Pareto set (%), the scatter of points in the space of solutions of the exterior set (X), and the scatter of points in the space of external criteria set (Y). All these criteria should be maximized.

The results of testing evolutionary algorithms for multicriteria optimization

		The best	Average	Worst
NSGAI	%	84	74	52
	X	0.97671259	0.947632	0.9093986
	Y	0.969613066	0.94658963	0.92338803
SPEA	%	93	70	54
	X	0.6079351	0.5958682	0.556933
	Y	0.719441	0.71654223	0.7102566
SPEA2	%	75	71	67
	X	0.7126218	0.5197434	0.309482
	Y	0.9603189	0.83283465	0.7125409

Table 2

Analysis of the test results shows:
 – SPEA 2 solves the problem better than SPEA in the sense of the second and third metrics;

- SPEA 2 spends more time on computations;
- NSGA II solves the problem better than SPEA and SPEA2: it gives minimal deviation from the Pareto set and a more uniform distribution of the obtained non-dominated solutions.

Thus, it was found that NSGA II with average mutation and uniform crossover is the most effective algorithm, i. e. the Pareto front includes the highest number of points from the external set, and also their variation in the spaces of alternatives and criteria is maximal.

2. Artificial Neural Networks

Genetic algorithm for adjusting of the ANN weight coefficients (GA-ANNW). The error back-propagation algorithm is one of the commonly used methods for training multilayer neural networks. This method has the following serious drawbacks:

- frequently converged into a local minimum;
- strong influence of the choice of the step size on the quality of the solution found.

In order to improve the accuracy of solutions it is possible to use a genetic algorithm for ANN training since it is effective for solving global optimization problems and could avoid the above-mentioned problems.

In this paper, the genetic algorithm was implemented to adjust the weight coefficients of fully connected multilayer feed-forward neural networks (GA-ANNW) [10]. We used the sigmoid as an activation function.

Weights are recorded sequentially in the chromosome as a binary code. An example of a chromosome is shown

in fig. 1, where 4 bits correspond to one weight coefficient. In solving real problems, the number of bits that are used to encode a single weighting coefficient depends on the accuracy of the settings and the spread of possible values of weights.

The effectiveness of GA-ANNW was tested on 14 test approximation problems and was found to be sufficiently high [10].

Genetic algorithm for automated design of artificial neural networks structures (GA-ANNS). During the design of ANN structure, the number of layers and the neurons on each layer must be determined and also the activation function type for each neuron must be established. Experts can identify an optimal ANN structure, but it is a time-consuming procedure. We propose the use of a genetic algorithm to automatically design the ANN structure (GA-ANNS) [10].

GA-ANNS uses a binary chromosome, whose example is shown in fig. 2. The corresponding neural network is presented in fig. 3. Hidden layers are coded sequentially. Each neuron is encoded in four bits. For each neuron, we firstly, randomly, with a fixed probability equal to 0.3, decided whether it will be used in the network or not. If in the network a neuron is not presented, its place in the chromosome is marked with zeros. Otherwise, it is randomly selected as one of the fifteen activation functions [11], whose number is written in binary code.

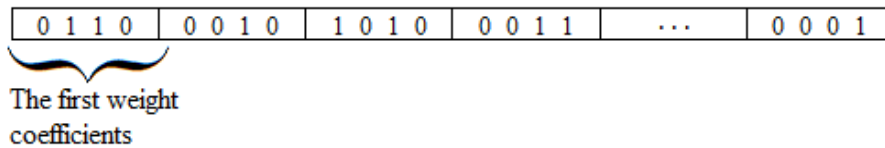


Fig. 1. Binary chromosome for ANN weight coefficients tuning

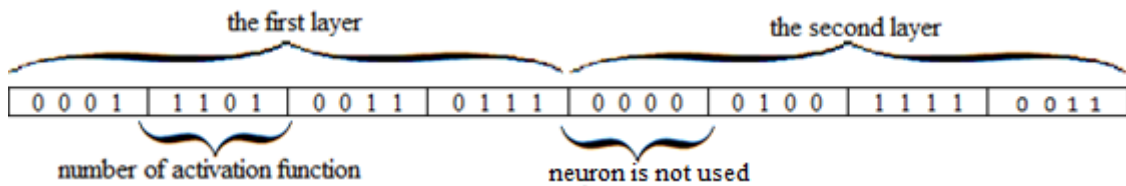


Fig. 2. Binary chromosome for GA-ANNS

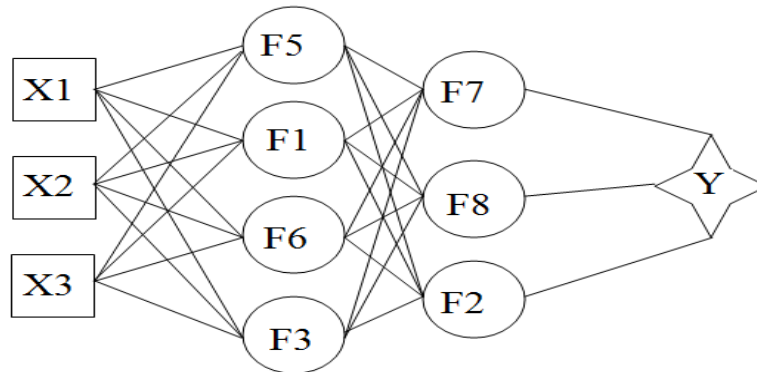


Fig. 3. Example of the ANN built using GA-ANNS

For each selected ANN structure GA-ANNW was run for the tuning of weighting coefficients.

Genetic algorithm with a choice of the most informative features (GA-ANNinput). Since the efficiency of the genetic algorithm for ANN depends on the dimension of the problem in hand, it is reasonable to avoid the use of uninformative features. The modification of the genetic algorithm for the choice of the most informative features during the automated design of ANN (GA-ANNinput) assumes the use of additional bits in the GA chromosomes. These bits determine whether an input is included in the input layer or not. The coding method can be seen in fig. 4, and the corresponding chromosomes for the neural network are shown in fig. 5.

Evolutionary algorithms for automated multicriteria design of ANN (MC-ANNinput) [12]. Solving the problem of the design of artificial neural networks, it is often difficult to find a compromise between the accuracy and simplicity of the solution. We propose to use evolutionary algorithms for this. We use an averaged modelling error, the ANN neuron number and the input number as

criteria for MC-ANNinput. The encoding of the binary string follows the same rules as in the previous method.

3. Performance Estimation of Evolutionary Algorithms for Automated Neural Network Design

The proposed algorithms were tested with the following rules:

- the performance for all genetic algorithm settings was estimated over 100 runs;
- the number of generations is equal to 1000;
- the number of individuals is equal to 500;
- the size of the training sample is equal to 70 % of the total number of examples, and the test sample size is equal to 30 %;
- results are presented for the best settings of the genetic algorithm;
- the maximum size of a network is equal to 5×5;
- error of all runs were averaged.

The effectiveness of the algorithms was estimated over 14 test problems of approximation from [7] and the results are presented in tab. 3.

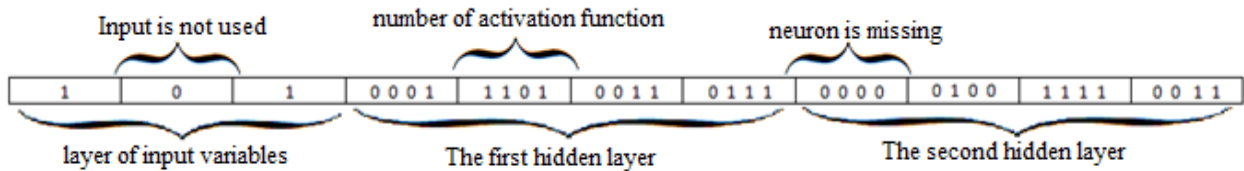


Fig. 4. Chromosome in GA-ANNinput

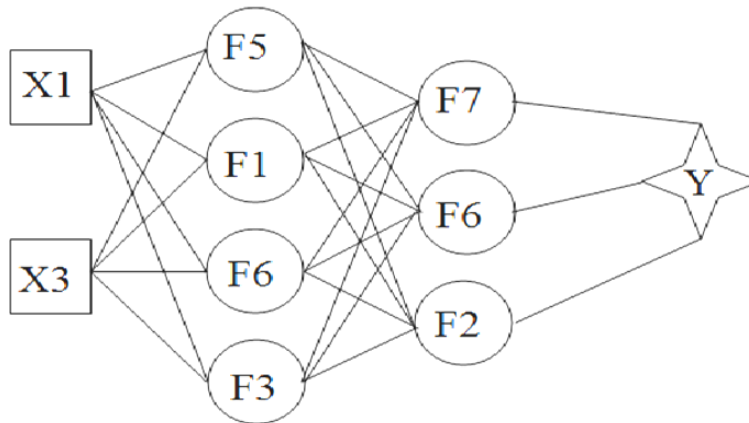


Fig. 5. Corresponding ANN

Table 3

Test results

Algorithm	Reliability	Generation	Average number of neurons
GA-ANNS	0.917 [0.315; 1]	15.7 [11; 29]	3.5
GA-ANNinput	0.896 [0.307; 1]	26.3 [31; 81]	3.4
GA-ANNW	0.543 [0.295; 0.987]	51.6 [42; 84]	25
MultiNSGAI-ANN	0.915 [0.279; 1]	19.9 [18; 48]	4
MultiSPEA2-ANN	0.874 [0.365; 0.953]	31.8 [34; 67]	4.7
MultiSPEA-ANN	0.832 [0.471; 0.921]	29.7 [38; 73]	6

The modification does not bring significant improvements, since the vector of input variables has a small dimension, so all inputs are informative. Therefore, it can be useful for tasks that are more complex.

The GA with proportional selection, two-point crossover and average mutation has achieved the highest accuracy. These settings should be used to solve real-world problems of data analysis.

4. Performance Estimation of Evolutionary Algorithms for Automated Neural Network Design on Real World Problems

Three real problems of data analysis [13] were used for the evaluation of the proposed algorithms: the Iris classification, Australian and German bank scoring problems. The parameters of these tasks (the number of inputs, the sample size, the number of classes and number of examples in each class) are shown in tab. 4. For solving these problems algorithms used 1500 generations and 750 individuals. All results were averaged over 100 runs.

The comparison of the implemented algorithms with other methods was performed for the Iris classification problem [14] and bank scoring problems [15]. The results

for the Iris classification problem and for bank scoring problems are shown in tab. 5 and in tab. 6 correspondingly.

As it can be seen from tab. 5, GA-ANNinput did not improve performance in comparison with GA-ANNS. The main reason for this is the same as for test problems (all inputs are sufficiently informative).

It can be seen from tab. 6 that the GA-ANNinput shows higher efficiency than GA-ANNS and MultiNSGAI-ANN. The modified algorithm takes second place among all the algorithms for the Australian problem, it loses only for the method specially developed for such tasks, and is in 6th place for the German problem. That is a good result for a non-specific for the problem method.

The algorithm uses such important data as a credit history or house ownership, but could reject less important data, like family status.

Fig. 6 and 7 show that the structures of the best neural networks are relatively simple. On average the developed algorithms used about 7 inputs (from 15) for the Australian credit problem and approximately 11 inputs (from 24) for the German credit problem and a network with 9 and 14 neurons, respectively, from the maximum possible 25 neurons.

Table 4

Numerical characteristics of the data analysis problems

Name of the task	Number of attributes	The sample size	Number of classes	Separation by class
Iris	4	150	3	Class 1: 50 Class 2: 50 Class 3: 50
Australian Credit Data	15	690	2	Class 1: 290 Class 2: 310
German Credit Data	24	1000	2	Class 1: 700 Class 2: 300

Table 5

The results for Iris classification problem

The method name	ES-ANN	CRO-ANN	EP-ANN	GSO-ANN	GA-ANNS	PSO-ANN	GA-ANNinput	MGNN	MultiNSGAI-ANN
Error	0.0066	0.0067	0.0116	0.0142	0.0201	0.0202	0.0231	0.0305	0.0312

Table 6

Performance comparison (classification errors)

The method name	Australian Credit Approval	German Credit Data
2SGP	0.0863	0.1985
GA-ANNinput	0.087	0.232
GA-ANNS	0.091	0.241
C4.5	0.1014	0.2227
MLP	0.1014	0.2382
MultiNSGAI-ANN	0.102	0.24
Fuzzy classifier	0.109	0.206
GP	0.1111	0.2166
k-NN	0.1256	0.2849
LR	0.1304	0.2163
Bayesian approach	0.153	0.321
Bagging	0.153	0.316
Boosting	0.24	0.3
CART	0.285	0.2435

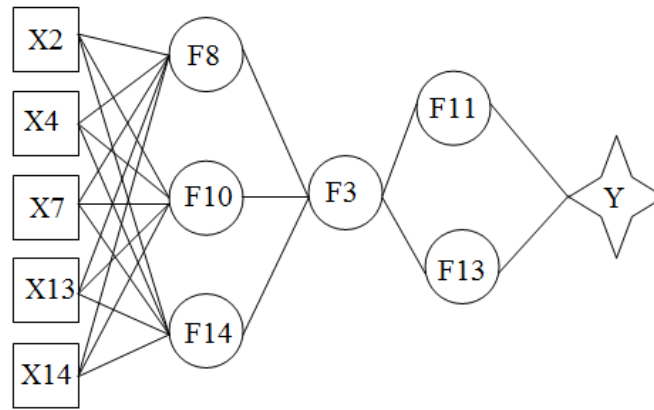


Fig. 6. The best found ANN for the Australian credit problem (GA-ANNinput)

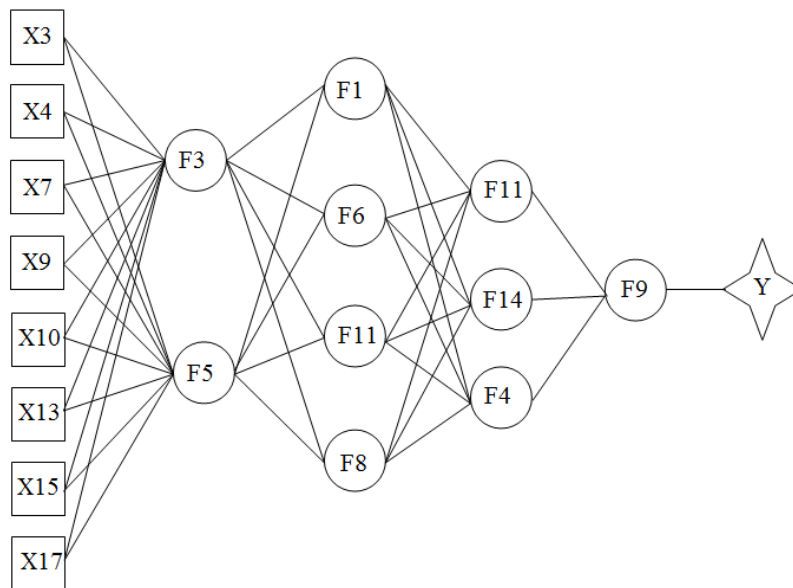


Fig. 7. The best found ANN for the German credit problem (GA-ANNinput)

For two bank scoring tasks statistical analysis of raw data was performed. It was found that these problems have a weak relationship between some input variables and output variables (class). Additionally, input attributes are divided into groups whose members are strongly correlated with each other and weakly correlated with members of other groups. Further analysis showed that the ANN obtained by the GA-ANNinput or MultiNSGA-ANN, as a rule:

- not used inputs that are weakly correlated with the output variable;
- select only significant inputs, one from each group and discard the rest.

At the same time, for problems in which all inputs are significant, GA-ANNinput typically includes input variables without losing information. We can conclude that the proposed tool, which allows not only the building of sufficiently effective neural network classifiers, but at the same time the establishment of the most significant features, is useful for further investigation.

Conclusion. In this paper, we tested a genetic algorithm for the tuning of weighting coefficients and ANN structure design for approximation problems. It is shown

that neural networks with a structure that is configured by using a genetic algorithm solve the problem of approximation on a good level compared to the classical methods.

A modified genetic algorithm with ANN structure design with a selection of the most informative features was developed and implemented. The modified algorithm for test approximation problems did not produce a significant improvement in comparison with the genetic algorithm without feature selection. This is due to the fact that the test problems input variables vector has a small dimension, and consequently the exclusion of any input is impossible and leads to a decrease in the solution accuracy.

However, for solving large-scale problems, the modified algorithm has shown higher accuracy, and taken fewer computational resources. The modified algorithm with determination of the most informative features together with the automated design of neural network classifiers includes as the input only variables of the original problem that are weakly correlated with one another.

In the future, we plan to develop a genetic algorithm for automated design of fuzzy logic systems with the simultaneous selection of the most informative features.

Acknowledgment. This work was supported by the Ministry of Education and Science of the Russian Federation, Project 140/14.

Благодарности. Работа поддержана Министерством образования и науки Российской Федерации, НИР 140/14.

References

1. Goldberg D. E. *Genetic algorithms in search, optimization and machine learning*. 1989, Reading, MA: Addison-Wesley, 403 p.
2. Kruglov V., Borisov V. [Artificial neural networks]. *Teoriya i praktika*. Moscow, Goryachaya liniya – Telekom Publ., 2002, 382 p. (In Russ.).
3. Girosi F., Jones M., Poggio T. Regularization theory and neural network architecture. *Neural Computation*. 1995, vol. 7, p. 219–270.
4. Seshadri A. *NSGA - II: A multi-objective optimization algorithm*. 2006, 524 p.
5. Wiley J. & Sons. *Multi-Objective Optimization Using Evolutionary Algorithms*. 2001, p. 497.
6. Holland J. H. *Adaptation in natural and artificial systems*. MI: University of Michigan Press, 1975.
7. Test problems unconditional one-criterion optimization. Available at: <http://dces.essex.ac.uk/staff/qzhang/moeacompetition09.htm>.
8. Luke S. *Essentials of Metaheuristics. A Set of Undergraduate Lecture Notes. Zeroth Edition*. 2009, Online Version 0.5.
9. Zhang Q., Zhou A., Zhao S., Suganthan P. N., Liu W., Tiwari S. Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition. *Proceedings of IEEE Congress on Evolutionary Computation (CEC'2009)*. 2009, p. 30.
10. Koromyslova A. A., Semenkin M. E. [Evolutionary design of neural network classifiers with the selection of the most informative features]. *Materialy konferentsii "Teoriya i praktika sistemnogo analiza"* [Proceedings of the conference "Theory and practice of system analysis"]. Part 2, p. 74–83 (In Russ.).
11. Haykin S. *Neural Networks and Learning Machines (3rd Edition)*. 2009, 906 p.
12. Brester Ch. Yu., Semenkin E. S. Development of adaptive genetic algorithms for neural network models multicriteria design. *Vestnik SibSAU*. 2013, No. 4(50), p. 99–103.
13. Frank A., Asuncion A. UCI Machine Learning Repository. 2010. *Irvine, CA: University of California, School of Information and Computer Science*. Available at: <http://archive.ics.uci.edu/ml>.
14. Yu J. J. Q., Lam A. Y. S., Li V. O. K.: Evolutionary Artificial Neural Network Based on Chemical Reaction Optimization. In: *IEEE Congress on Evolutionary Computation (CEC'2011)*, 2011, pp. 2083–2090.
15. Sergienko R., Semenkin E., Bukhtoyarov V. Michigan and Pittsburgh Methods Combining for Fuzzy Classifier Generating with Coevolutionary Algorithm for Strategy Adaptation. *Proceedings of IEEE Congress on Evolutionary Computation (CEC'2011)*. 2011, p. 113–120.

Библиографические ссылки

1. Goldberg D. E. *Genetic algorithms in search, optimization and machine learning*. Reading, MA : Addison-Wesley, 1989. 403 p.
2. Круглов В. В., Борисов В. В. Искусственные нейронные сети // Теория и практика. 2-е изд. М. : Горячая линия – Телеком. 2002. 382 с.
3. Girosi F., Jones M., Poggio T. Regularization theory and neural network architecture // *Neural Computation*. 1995. Vol. 7. Pp. 219–270.
4. Seshadri A. *NSGA - II: A multi-objective optimization algorithm*. 2006. 524 p.
5. Wiley J. & Sons. *Multi-Objective Optimization Using Evolutionary Algorithms*. 2001. Pp. 497.
6. Holland J. H. *Adaptation in natural and artificial systems*. MI : University of Michigan Press, 1975.
7. Test problems unconditional one-criterion optimization. URL: <http://dces.essex.ac.uk/staff/qzhang/moeacompetition09.htm>.
8. Luke S. *Essentials of Metaheuristics // A Set of Undergraduate Lecture Notes / Zeroth Edition*. Online Version 0.5. 2009.
9. Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition / Q. Zhang [et al.] // *Proceedings of IEEE Congress on Evolutionary Computation (CEC'2009)*. 2009. Pp. 30.
10. Коромыслова А. А., Семенкина М. Е. Эволюционное проектирование нейросетевых классификаторов с выбором наиболее информативных признаков // Теория и практика системного анализа : материалы конференции. Ч. 2. С. 74–83.
11. Haykin S. *Neural Networks and Learning Machines*. 3rd Edition. 2009. 906 p.
12. Brester Ch. Yu., Semenkin E. S. Development of adaptive genetic algorithms for neural network models multicriteria design // *Вестник СибГАУ*. 2013. Вып. 4 (50). С. 99–103.
13. Frank A., Asuncion A. UCI Machine Learning Repository. Irvine, CA : University of California, School of Information and Computer Science, 2010. URL: <http://archive.ics.uci.edu/ml>.
14. Yu J. J. Q., Lam A. Y. S., Li V. O. K.: Evolutionary Artificial Neural Network Based on Chemical Reaction Optimization. In: *IEEE Congress on Evolutionary Computation (CEC'2011)*, 2011, p. 2083–2090.
15. Sergienko R., Semenkin E., Bukhtoyarov V. Michigan and Pittsburgh Methods Combining for Fuzzy Classifier Generating with Coevolutionary Algorithm for Strategy Adaptation // *Proceedings of IEEE Congress on Evolutionary Computation (CEC'2011)*. 2011. P. 113–120.