If we compare two schemes, the sum of cases when algorithm founded global optimum point for different settings is 512 and 476 for algorithm with changing of expected value and not, respectively. The number of total runs was 796 and 819, the number of populations that was aborted because of their best solution being close to one from the set was 534 and 516, respectively. The last fact means that the checking the distance between point that was already suspected to be «final» improves the performance as well and prevent from extra evaluations. On the fig. 4 the relation between increasing of the tail size and number of restarts for different boarders is on the left diagram, and number of global optimum points found for different boarder values and increasing of the tail size is on the right diagram.As it can be seen on the figures, there is nonlinear influence of increasing the size of the tail, but the size of the tail does change the algorithm efficiency, as well as the boarder size. The further study will be focused on different schemes of critique's action and detection and ways to adapt the new parameters. Anyway, even now, with the same number of function evaluation we increased the estimated probability to find the desired solution from 0.65 to 1.

References

1. Popescu M., Dumitrache M. 2005: On the optimal control of affine nonlinear systems. Mathematical Problems in Engineering (4): p. 465–475.
2. Chen W. H., Balance D. J., Gawthrop P. J., 2003: Optimal control of nonlinear systems: a predictive control approach. Automatica 39(4): p. 633–641.
3. Primbs J. A., Nevistic V., Doyle J. C., 1999: Nonlinear optimal control: a control Lyapunov function and receding horizon perspective. Asian Journal of Control, vol. 1, no. 1: p. 14–24.
4. Rao A. V., 2009: Survey of Numerical methods for Optimal Control, AAS/AIAA Astrodynamics Specialist Conference, AAS Paper 09-334, Pittsburg, PA.
5. Bertolazzi E., Biral F., Lio M., 2005: Symbolic-Numeric Indirect Method for Solving Optimal Control Problems for Large Multibody Systems: The Time-Optimal Racing Vehicle Example. Multibody System Dynamics, vol. 13, no. 2, p. 233–252.
6. Cruz P., Torres D., 2007, Evolution strategies in optimization problems, Proc. Estonian Acad. Sci. Phys. Math., 56, 4, p. 299–309
7. Kirk D. E., 1970: Optimal Control Theory: An Introduction. Englewood Cliffs, NJ: Prentice-Hall.
8. Schwefel H.-P., 1995: Evolution and Optimum Seeking. New York: Wiley & Sons.
9. Hansen N., 2006: The CMA evolution strategy: a comparing review. Towards a new evolutionary computation. Advances on estimation of distribution algorithms, Springer, p. 1769–1776.
10. Storn R., Price K., 1997. Differential Evolution – A Simple and E?cient Heuristic for Global Optimization over Continuous Spaces, Journal of Global Optimization, 11, p. 341–359.
11. Kennedy, J., Eberhart, R., 2001. Swarm Intelligence. Morgan Kaufmann Publishers, Inc., San Francisco, CA.

УДК 519.6

INTELLIGENT INFORMATION TECHNOLOGIES IN TIME SERIES FORECASTING

E. S. Semenkin, A. A. Shabalov

Siberian State Aerospace University named after academician M. F. Reshetnev
31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660014, Russian Federation
E-mail: eugenesemenkin@yandex.ru, shabalov-andrey@mail.ru

Intelligent information technologies enable to solve complex data mining problems in various domains of human activity. In this paper such popular techniques as artificial neural networks, fuzzy rule based systems and neuro-fuzzy systems are considered. A genetic programming algorithm is used for building intelligent systems ensembles in order to improve the performance and reliability of decision making. The methods proposed are applied to time series prediction task. The results obtained are compared to other state-of-the-art time series forecasting techniques.

Keywords: artificial neural networks, fuzzy rule based systems, neuro-fuzzy systems, evolutionary algorithms, ensembles of intelligent systems.

ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ПРОГНОЗИРОВАНИИ ВРЕМЕННЫХ РЯДОВ

Е. С. Семенкин, А. А. Шабалов

Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева
Российская Федерация, 660014, Красноярск, просп. им. газ. «Красноярский рабочий», 31
E-mail: eugenesemenkin@yandex.ru, shabalov-andrey@mail.ru

Интеллектуальные информационные технологии способны решать сложные задачи интеллектуального анализа данных в различных областях деятельности человека. В данной статье рассматриваются такие популярные инструменты, как искусственные нейронные сети и нейро-нечеткие системы. Алгоритм генетического программирования используется для построения ансамблей интеллектуальных информационных технологий в целях улучшения эффективности и надежности принятия решений. Предлагаемые методы апробированы на задачах прогнозирования временных рядов. Представленные результаты сравнены с другими распространенными алгоритмами прогнозирования временных рядов.

Ключевые слова: искусственные нейронные сети, системы на нечеткой логике, нейро-нечеткие системы, эволюционные алгоритмы, ансамбли интеллектуальных систем.

In order to control and design complex systems one has to have a model of an object (process). However, real complex system modeling is a difficult task. A simulation model can be a solution of the problem (computer simulation model of the system/object). In practice as a rule, there is a big amount of raw data of observations of the system behavior. Intelligent information technologies (IIT) enable to obtain a simulation model on short time. Having such a model it becomes possible to examine and track the properties of the simulated system what allows developing finite system model at a later date.

Intelligent systems have got a wide propagation in different fields of human activity connected with complex system modeling and optimization tasks. Evolutionary algorithms [1], fuzzy rule based systems [2], artificial neural networks [3] and neuro-fuzzy systems [4] and other techniques and technologies are of a popular school for investigation among scientists of this domain. These tools make it possible solving complex intelligent problems which are difficult to solve, or practically impossible, with classic techniques [5].

Along with single technologies, hybrid approaches are developed. Hybridization of neural networks and evolutionary algorithms (EA), fuzzy rule based systems and EA and neural networks and fuzzy systems have resulted in substantial growth of investigation in intelligent system design domain.

However, design of intelligent information technologies is a complex optimization problem whose structure considerably impedes applying of classic techniques. Moreover, solving such a problem requires substantial financial expenditure and time costs.

Genetic algorithms (GA) represent a stochastic optimization procedure based on evolution and natural selection principle. GAs have demonstrated high performance in solving practical multiextremal problem [6, 7]. Flexible parameter coding structure of a genetic algorithm enables effective applying for IIT structure design as well as tuning their parameters [8].

At the present time by virtue of computing power gain ensemble approaches become more popular in different approximation and classification tasks. It has been observed that heterogeneity of the ensemble members plays an important role in building up a terminal decision [9]. Different approaches have been proposed to maintain heterogeneity of the ensemble members. Among them, running on different feature sets [10], training sets (bagging [11] and boosting [12]). The diversity of the ensemble can be reached as well by generation of different member structures. For instance, generation of neural networks of different structures by running on the same training and feature sets. In order to compute the ensemble output, commonly, simple and weighted averaging are used In classification task along with aforementioned methods ranking and majority voting are used as well [13; 14].

In [15] Ramírez et. al. used Mamdani fuzzy inference system to combine outputs of several techniques (Fuzzy KNN, Multi Layer Perceptron with Gradient Descent with Momentum Backpropagation, and Multi Layer Perceptron with Scaled Conjugate Gradient Backpropagation). A genetic algorithm was applied for selection definite neural networks from pre-generated set according to the performance metrics [16]. Siwek et. al. [17] used 4 neural-like predictors (Multilayer Perceptrons (MLP), Support Vector Machines (SVM), Elman Networks, and Radial Basis Functions Networks). The obtained results were post-processed by SVM or MLP. Johansson et. al. [18] used a genetic programming method for building an ensemble from predefined number of Artificial Neural Networks. Functional set of a genetic programming algorithm consisted of averaging and multiplying and terminal set included generated neural networks models and constants.

In all abovementioned examples ensemble member structures were generated by hand by trail-and-error method.

A genetic programming algorithm [19] operates by computer programs expressed by trees structures (as a rule, by binary trees). The operation of the algorithm is similar to a genetic algorithm described above. Before the start of the running the algorithm it is necessarily to specify a functional set (collection of functions used) and a terminal set (collection of system variables, collection of constants used).

In this paper we consider applying a genetic programming algorithm for intelligent information technologies ensemble design. As opposed to Johansson et. al. work a terminal set is presented by an extended collection of elementary functions. Another peculiarity of our work consists in applying diverse intelligent systems providing by that heterogeneity of the ensemble. Moreover, neural networks, fuzzy rule based systems and neuro-fuzzy systems are generated automatically on the basis of self-adapting genetic algorithms what allows to skip expensive involvement of experts.

The article is organized as follows. In Section I the description of IIT algorithmic core generation automated methods is given. In Section II the description of IIT ensemble design procedure by means of genetic programming algorithm is presented. Numerical experiments and performance comparing with other up-to-date techniques on time series forecasting problems are given in Section 3. In Conclusion the results of the work done and future direction of investigations are discussed.

Automated design of intelligent information technologies algorithmic core. *Artificial neural networks*. In the work a multilayer perceptron in the capacity of architecture structure of a neural network was taken as being widely spread in practical applications. While designing the architecture of a neural network the following problems occur. The choice of an architecture structure (number of hidden layers and number of hidden neurons on each hidden layer). As a rule for tuning of weights coefficients of such networks a back-propagation algorithm and its different modifications are used [20–22] which are based on gradient descent method. The drawback of such algorithms consist in: low convergence speed, noise sensitivity, algorithm performance dependency on learning heuristic step, and, as a rule, modeling error does not reach the global optimum due to function complexity [23].

To overcome such problems it is suggested to apply genetic algorithms for neural network structure generation as well as weights coefficients tuning. The detailed description of the algorithm scheme and the way of parameters coding can be found in [24].

Fuzzy rule based systems. While developing a fuzzy system an expert faces the problem of initial fuzzy rules selection a set of which could be incomplete and contradictory. While developing a fuzzy system an expert faces the problem of initial fuzzy rules selection a set of which could be incomplete and contradictory. The selection of membership functions parameters describing the input and output object parameters is carried out subjectively and may represent the reality incorrectly. Moreover, fuzzy logic systems don not have automatic learning algorithms.

Taking this into account, to improve decision making validity the genetic algorithms were applied. When designing a fuzzy system structure a Pittsburgh approach was used [25] in which single individual represents the whole rule base. The realized coding scheme of fuzzy system parameters enables to determine automatically the size of a rule base, i.e. the number of rules, as well as the length of each single rule, i. e. the number of input parameters in left part of a rule, due to the inclusion of an additional term – "don't care" term [26]. The parameter coding schemes can found in [27].

Neuro-fuzzy systems. The generation process of neuro-fuzzy systems consists of two phases [28; 29]. The first stage (unsupervised mode) represents the initial numerical data clustering. After that the coarse fuzzy rules are determined. The second stage (supervised mode) consists in accurate tuning of the rule base derived. Usually gradient algorithms are used here the drawbacks of which are widely known and prevent effective use of neuro-fuzzy systems. Therefore, for membership functions parameters tuning the GAs were applied instead of gradient algorithms. Their performance was shown in previous works and outperformed the performance of the steepest descent algorithm in practical problems solving in terms of modeling relative error [30]. The parameters coding scheme of neuro-fuzzy systems into a genetic algorithm strand are described in [31].

Self-adapting genetic algorithm. For intelligent information technologies structure generation and their parameter tuning a self-adapting genetic algorithm was developed based on asymptotic genetic algorithm [32]. This algorithm operates by probability distribution vector of 0 or 1 bit occurrence in respective chromosome gene. On the basis of asymptotic selection and asymptotic mutation with adaptive setting of mutation probability value [33] the following customized parameters left: type of selection, (not)applying elitism strategy. The crossover operator in explicit form is absent. The selection automation of parameters left allows to facilitate the work to a user being not an expert in evolutionary calculation domain.

The process of automatic selection of a selection type in self-adapting asymptotic genetic algorithm is carried out automatically dynamically in the course of algorithm running on the basis of parameters probabilistic mixture. Let z_k be a probability of k-th selection type applying. On every generation the probabilities are recalculated based on the following formula (in order to prevent probabilities approaching close to zero 20 percent of probability is divided equally among every parameter value):

$$z_k = z_{all} + \frac{r_k \cdot \left(100 - N \cdot z_{all}\right)}{\sum_{k=0}^{K} r_k},$$

where K – number of values of tunable parameter; $k = \overline{1, K}$, $z_{all} = \frac{20}{K}$ $r_k = \frac{success_k^2}{used_k}$ – ratio, where $used_k$ – number of times when k-th operator was applied; $success_k$ – number of times of k-th operator which led to average fitness improvement of a population comparing to previous generation. Initially $used_k$ are set to 1 in order to avoid the division by zero. The scheme of this GA is similar to the asymptotic GA with the additional step of probability distribution vector recalculation of selection type [24].

The proposed techniques of IIT algorithmic core generation were successfully applied to different real-world problems solving. For conducting such experiments a program system π-IT-on was developed [34; 35]. In table 1 the list problems solved is presented. Part of them was taken from machine learning repository UCI [36].

Problems 1, 2 and 4 are of classification tasks. The rest are of approximation tasks. For every problem 20 runs were implemented for every IIT type generation. In table 2 the best results are given in terms of relative error criterion. In the table the following notations are used: Tr – the error on a training set, Ts – the error on a test set. From the table one can see that in most cases neuro-fuzzy systems outperformed other technologies. The performance of all realized intelligent systems is comparable to known results.

Table 1

Characteristics of real-world problems

	Problem	Input dimension	Output dimension	Sample size	
				Training	Test
	Machine learning repository UCI				
1.	Iris classification	4	3	135	15
2.	Wine classification	13	3	163	15
3.	Forest fires forecasting	12	1	477	40
4.	Satellite image classification	36	6	4435	2000
	Applied problems				
5.	Turbine condition monitoring based on forecasting of vibration signals	11	12	1000	400
6.	Ore-thermal process modeling	9	1	47	10
7.	The degradation prediction of electrical characteristics of spacecraft's solar arrays	7	4	177	20
8.	Test-based characteristics forecasting of jet engine	5	1	20371	2263

Table 2

The results of real-world problem solving

№	Нейронная сеть		Система на нечеткой логике			Нейро-нечеткая система		
	Error		Error		Rule number	Error		Rule number
	Tr, %	Ts, %	Tr, %	Ts, %		Tr, %	Ts, %	
1	3,70	6,66	1,48	0	5	1,48	0	3
2	0,61	6,66	0	0	7	0	0	5
3	1,78	1,79	1,11	1,11	5	1,45	1,46	4
4	23,2	24,3	16,87	19,61	15	15,67	17,5	9
5	9,11	9,14	8,07	8,09	15	7,99	7,97	10
6	4,86	4,97	2,99	3,01	15	2,81	2,92	10
7	9,01	9,72	5,66	7,66	17	5,05	5,87	15
8	8,29	8,73	4,97	5,01	24	0,93	0,95	20

Evolutionary approach of intelligent information technologies ensemble design. In the majority of cases real-world problems are large-scale and complex for solving by a single technology. Ensembles of intelligent systems allow to incorporate different technologies for resultant decision making what enables to improve the performance and reliability of a terminal system.

In the work for effectiveness and reliability improvement of IIT it is suggested to apply the genetic programming method in order to form both IIT ensemble composition for complex problems solving and the way of cooperation of ensemble members in making the resultant decision based on particular decisions of individual technologies.

The resultant solution is comprised of mathematical expression from individual decisions of generated intelligent systems. Thus, partial decision of single technologies will be terminal set elements.

On a preliminary stage scheme it is necessary to generate and train in advance the specified number of terminal set elements which later will be used in the algorithm. In this scheme, there exist two modes of mutation realization in the genetic programming algorithm. It is possible either to choose randomly an element from the terminal set or to generate an absolutely new intelligent system. A functional set includes mathematical expressions.

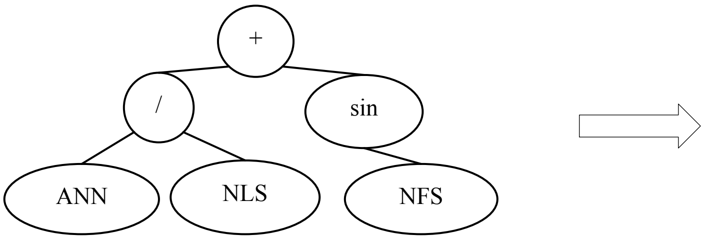
Thus, combination of individual technologies in the IIT ensemble enables to integrate the advantages of every of them and considerably to compensate their drawbacks improving in such a way the performance and reliability of the system in a whole.

There are the examples of tree coding in the genetic programming algorithm below. On fig. 1 an example of a tree genotype (on the left) and its correspondent decision in the search space is presented. The following notations are used: ANN – artificial neural network, FLS – fuzzy logic system, NFS – neuro-fuzzy system.

For described earlier list of real-world problems in Section 1 correspondent ensembles were generated. In order to build an ensemble preliminarily 10 intelligent systems of every type were generated. For instance, for ore-thermal process modeling the following formula was obtained:

$$Ni(\%) = NFS_{10} \cdot e^{\frac{FLS_6 \cdot e^{\frac{FLS_6}{NFS_9}}}{FLS_{10}}}$$

The relative error was equal to 2,21 % on the training set and 2,33 % on the test set what is better than for every individual IIT.



$$OUT = \frac{ANN}{NLS} + \sin(NFS)$$

Fig. 1. Genotype and phenotype representations

In wine classification problem the following expression was got:

$$C = \sin\left(NFS_4 \cdot \sqrt{e^{NFS_{10}}}\right),$$

where C is the class number. A recognition error constituted 0 % on both training and test sets.

In table 3 a comparison with other up-to-date methods of ensemble building for the Iris classification problem is given [37]. The proposed techniques are highlighted in bold. From the table it can be seen that the ensemble allows to reach hundred-per-cent successful classification.

Table 3

Comparison with analogs

Classifiers	Error, %
Ensemble (ANN+FLS+NFS)	**0,00**
CROANN	1,31
SVM-best	1,40
GSOANN	3,52
NFS (weighted average)	**4,11**
NFS (simple average)	**4,33**
CCSS	4,40
NLS (weighted average)	**5,06**
NLS (simple average)	**5,33**
ANN (weighted average)	**5,37**
ANN (simple average)	**5,66**
GANet-best	6,40
ESANN	7,08
PSOANN	10,38
EPANN	12,56
SGAANN	14,20

Experimental investigation of time series forecasting problems solving. For testing of proposed IIT design algorithms on time series forecasting problems sets of data were used taken from “Synthetic Control Chart Time Series Data Set” from machine learning repository UCI [36].

These samples are synthetic tests for prediction algorithms. Four classes of time series were used for testing: normal (1), cyclic (2), increasing trend (3) and decreasing trend (4). Solving different time series types in test problems allows to estimate well the capabilities of forecasting algorithms.

Every collection contains 60 values. 57 training tuples were used to generate an ensemble. Thus, for values $x(t)$, $x(t-1)$ and $x(t-2)$ it is necessary to predict $x(t+1)$. 20 independent runs of the program were implemented. In table 4 the results obtained compared to other methods are given [38] based on average relative error calculated as follows:

$$ERROR = \frac{100\ \%}{s\left(y_{\max} - y_{\min}\right)} \sum_{i=1}^{s} \left|o_i - y_i\right|,$$

where s – the number of predicted values; y_{max} and y_{min} – maximum and minimum observed values of a time series accordingly; y_i – true value of a time series, o_i – model output.

From given table one can see that the IIT ensemble always allows to improve the performance of a resultant system. Moreover, in every case it turned out to be the best from compared techniques. Realized fuzzy rule based systems and neuro-fuzzy systems generated automatically by means of genetic algorithms proved to be better than ensemble techniques GASEN and PGNS and GPEN. Exponential smoothing has demonstrated the worst modeling quality of time series.

Conclusions. In this work the algorithms of intelligent information technologies automated design on the basis of evolutionary algorithms were considered. The algorithmic core design of neural network models, fuzzy logic systems and neuro-fuzzy systems is carried out by the means of self-adapting genetic algorithm enabling to reduce to minimum the participation of an expert.

It is shown that forming the ensemble based on partial decisions of single technologies allows to improve the performance and reliability of a resultant system.

The effectiveness of applying developed algorithms in approximation and classification tasks is shown. The perspective of proposed approaches in time series forecasting problem solving has been demonstrated.

The future work is aimed on conducting additional experiments in time series forecasting problems solving, solving other real-world problems, comparison with up-to-date data mining techniques.

Table 4

Time series test

Method	Error, %			
	(1)	(2)	(3)	(4)
Ensemble (ANN+FLS+NFS)	**2,0**	**1,9**	**2,2**	**1,9**
ANN (simple average)	**22,1**	**12,1**	**14,6**	**8,1**
NLS (simple average)	**3,6**	**3,5**	**3,3**	**2,2**
NFS (simple average)	**3,2**	**2,8**	**3,1**	**2,5**
GASEN	11,3	9,7	10,8	9,6
Exponential smoothing	19,9	29,5	19,4	18,6
PGNS and GPEN	8	6,9	8,4	7,3

References

1. Eiben A. E., Smith J. E. Introduction to evolutionary computing. Springer, Berlin, Germany, 2003.

2. Yager R. R., Filev D. P. Essentials of fuzzy modeling and control. Wiley, New York, USA, 1994.

3. Rojas R. Neural networks: a systematic introduction. Springer, Berlin, Germany, 1996.

4. Ross T. J. Fuzzy logic with engineering applications. Wiley, England, 2004.

5. Konar A. Computational Intelligence: Principles, techniques and applications. Springer, Berlin, Germany, 2005.

6. Haupt R. L., Haupt S. E. Practical Genetic Algorithms. Wiley-Interscience, New Jersey, USA, 2004.

7. Goldberg D. E. Genetic Algorithms in Search, Optimization and Machine Learning. Ad-dison-Wesley Longman Publishing Co., Boston, MA, USA, 1989.

8. Semenkin E., Shabalov A., Efimov S. An automated design of intelligent information technologies ensembles by a genetic programming method. *Vestnik SibGAU*. 2011, № 3 (36), p. 77–81. (in Russian)

9. Dietterich T. G. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Machine Learning*, 2000, vol. 40, no. 2, p. 139–158.

10. Ho T. K., Hull J. J. and Srihari S. N. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1994, vol. 16, no. 1, p. 66–75.

11. Breiman L. Bagging predictors. *Machine Learning*. 1996, vol. 24 (2), p. 123–140,

12. Friedman J. H., Hastie T., Tibshirani R. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*. 2000, vol. 28, no. 2, p. 337–374.

13. Navone H. D., Granitto P. M., Verdes P. F., Ceccatto H. A. A learning algorithm for neural network ensembles. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*. 2001, no. 12, p. 70–74.

14. Kittler J., Hatef M., Duin R. P. W., Matas J. On Combining Classifiers, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20:3 1998.

15. Ramírez E., Castillo O., Soria J. Hybrid System for Cardiac Arrhythmia Classification with Fuzzy K-Nearest Neighbors and Multi Layer Perceptrons combined by a Fuzzy Inference System, WCCI 2010 IEEE World Congress on Computational Intelligence, Barcelona, Spain, 2010.

16. Amorim Neto M. C., Tavares G., Alves V. M. O., Cavalcanti G. D. C. and Ing Ren T. Improving Financial Time Series Prediction Using Exogenous Series and Neural Networks Committees, WCCI 2010 IEEE World Congress on Computational Intelligence, Barcelona, Spain, 2010.

17. Siwek K., Osowski S. and Sowinski M. Neural predictor ensemble for accurate forecasting of PM10 pollution, WCCI 2010 IEEE World Congress on Computational Intelligence, Barcelona, Spain, 2010.

18. Johansson U., Lofstrom T., Konig R., Niklasson L. Building Neural Network Ensembles using Genetic Programming, IJCNN '06. International Joint Conference on Neural Networks, 2006.

19. Koza J. R., Genetic programming. The MIT Press, London, England, 1998.

20. Hristev R. M. The ANN Book, 1998.

21. Kröse B., Smagt P. An Introduction to Neural Networks, 1996.

22. Wasserman P. Neural Computing. Theory and Practice. Moscow, Mir, 1984. (in Russian)

23. Kasabov N. Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering. The MIT Press, Cambridge, Second printing, 1998.

24. Sabalov A. Automated design of intelligent information technologies ensembles in data mining problems solving. *Theory and practice of system analysis: Proceedings of II All-Russian scientific conference of young scientists*. Rybinsk: RSATA named after P. A. Solovyev, 2012, p. 69–79. (in Russian)

25. Herrera F. and Magdalena L. Genetic Fuzzy Systems: a Tutorial. CICYT, 1995.

26. Ishibuchi H., Nojima Y. Analysis of interpretability-accuracy trade-off of fuzzy systems by multiobjective fuzzy genetics-based machine learning. *International Journal of Approximate Reasoning*. 2007, vol. 44, no. 1, p. 4–31.

27. Shabalov A., Semenkin E., Galushin P. Integration of Intelligent Information Technologies Ensembles for Modeling and Classification. Hybrid Artificial Intelligence Systems. Lecture Notes in Computer Science, Volume 7208/2012, p. 365–374.

28. Castellano G., Fanelli A.M. A self-organizing neural fuzzy inference network. Proceed-ings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. Como, Italy, 2000, vol. 5, p. 14–19.

29. Castellano G., Fanelli A. M. Information granulation via neural network based learning. IFSA World Congress and 20th NAFIPS International Conference. Vancouver, Canada, 2001, vol. 5, p. 3059–3064.

30. Shabalov A., Semenkin E., Efimov S. Ensembles techniques in data mining. Saarbruecken: LAMBERT Academic Publishing, 2012, 100 p. (in Russian)

31. Shabalov A., Semenkin E., Galushin P. Automatized Design Application Of Intelligent Information Technologies for Data Mining Problems. Joint IEEE Conference "The 7th Inter-national Conference on Natural Computation & The 8th International Conference on Fuzzy Systems and Knowledge Discovery", Shanghai, China, p. 2659–2662 (2011).

32. Galushin P., Semenkina O, Shabalov A. Comparative analysis of two distribution building optimization algorithms. 9th International Symposium on Distributed Computing and Artificial Intelligence, Salamanca, Spain, 2012.

33. Vorozheykyn A., Semenkin E. Development and investigation of adaptive probabilistic genetic algorithm for multi-criterion conditional optimization. Proceedings of international theoretical and practical conferences "Intelligent systems" (AIS'08) and (Intelligent CAD) (CAD'08). Moscow, Fizmatlit, 2008, no. 1, p. 15–21. (in Russian)

34. Semenkin E., Shabalov A V. An automated design system of intelligent information technologies ensembles. Program product and systems. 2012, № 4 (100), p. 51–54. (in Russian)

35. Semenkin E., Shabalov A. Program system π-IT-on for an automated design of intelligent information technologies ensembles. XIII national conference on the artificial intelligence with international participation CAI-2012: conference proceedings. Vol. 4. Belgorod: Publishing house BSTU, 2012, p. 109–116. (in Russian)

36. Frank A., Asuncion A. (2010). UCI Machine Learning Repository. Available at: http://archive.ics.uci.edu/ml.

37. 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), New Orleans, LA, 2011.

38. Bukhtoyarov V., Semenkin E., Sergienko R. Evolutionary approach for automatic design of neural networks ensembles for modeling and time series forecasting, Iadis international conference: Intelligent systems and Agents, MCCSIS, 2011.

УДК 519.8

SELF-CONFIGURING EVOLUTIONARY ALGORITHMS FOR TRAVELLING SALESMAN PROBLEM

O. E. Semenkina, E. A. Popov, O. E. Semenkina

Siberian State Aerospace University named after academician M. F. Reshetnev
31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660014, Russian Federation
E-mail: oleese@mail.ru, epopov@bmail.ru, semenkina.olga@mail.ru

This paper considers genetic algorithm (GA) and ant colony optimization algorithm (ACO) with the automated choice of operators for the travelling salesman problem solving. The choice is based on operator probabilistic rates calculated during algorithm execution. The performance comparison with other heuristics such as Lin-Kernigan heuristic (3-opt) and Intelligent Water Drops algorithm (IWDs) is fulfilled and competitive results are demonstrated.

Keywords: genetic algorithm, travelling salesman problem, ant colony algorithm.

САМОКОНФИГУРИРУЮЩИЙСЯ ЭВОЛЮЦИОННЫЙ АЛГОРИТМ ДЛЯ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА

О. Е. Семенкина, Е. А. Попов, О. Э. Семенкина

Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева
Российская Федерация, 660014, Красноярск, просп. им. газ. «Красноярский рабочий», 31
E-mail: oleese@mail.ru, epopov@bmail.ru, semenkina.olga@mail.ru

Рассматриваются генетический алгоритм (ГА) и алгоритм оптимизации на основе муравьиных колоний с автоматическим выбором операторов для решения задачи коммивояжера. Выбор основан на вероятностном ранжировании операторов в течение работы алгоритма. Представлено сравнение эффективности с другими алгоритмами, такими как алгоритм Лин-Кернигана и алгоритм интеллектуальных водяных капель, показаны соответствующие численные результаты.

Ключевые слова: генетический алгоритм, задача коммивояжера, алгоритм муравьиных колоний.