

**Distributed computing.** EA realization by means of distributed computing application is suggested. It means that all node calculations shall be executed by logic cores of computing systems, while algorithm itself shall not be altered. The main idea is that each one of the threads processes certain groups of individuals.

Worth noting that EA are stochastic. Distribute computing shortens the time-consumption without affecting any other criteria. Such effect has been achieved by means of master-thread usage, which was responsible for random number generator. Wilcoxon signed-rank test as well proves that reliability has not been affected.

Test has been conducted on a PC with following specifications:

- Operating System: Microsoft Windows 7;
- CPU: AMD FX8320 (@3.5GHz);
- RAM: 16GB.

Following graph shows dependency of the time-consumption from a number of logic cores used.

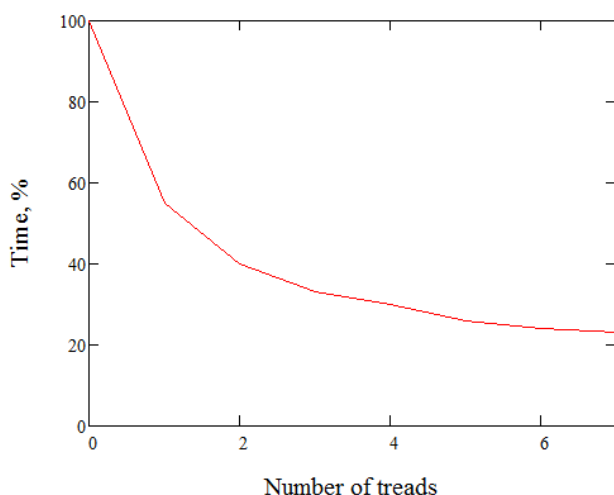


Fig. 6. The test results

**Conclusions.** Following algorithms have been developed and implemented: distributed self-configuring GA and GPA. Algorithms have been tested on a series of tasks and proved their effectiveness. Additionally, on ba-

sis of these algorithms the new artificial neural network design systems have been developed and tested on classification tasks.

Self-configuration enables to solve one of the actual tasks – genetic operator selection in EA. Distributed computing shortens the time-consumption without affecting efficiency. It might be useful in the case that involves great amount of time being spent on calculation. Judging from presented approaches comparison one should conclude that GPA is preferable in cases of artificial neural network design.

## References

1. Holland J. H. Adaptation in Natural and Artificial Systems, Ann Arbor: University of Michigan Press, 1975.
2. Holland J. H. Adaptation in natural and artificial systems. MIT Press. Cambridge, VA, 1992.
3. Rutkovskaya D., Pilinsky M., Rutkovsky L. Neural networks, genetic algorithms and fuzzy systems: translation from polish. I. D. Rudinsky. Moscow, Hotline. Telecom, 2006, 383 p.
4. Myasnikov A. S. Insular genetic algorithm with genetic operators probabilities dynamic choice. Available at: <http://technomag.edu.ru/doc/136503.html>.
5. Koza J. Genetic Programming. On the Programming of Computers by Means of Nature Selection, 1992.
6. Semenkina E., Semenkina M. Self-Configuring Genetic Programming Algorithm with Modified Uniform Crossover. In: Proc. of IEEE Congress on Evolutionary Computation. IEEE World Congress on Computational Intelligence, Brisbane, Australia, 2012.
7. Poli R., Langdon W. B. On the Ability to Search the Space of Programs of Standard, One-Point and Uniform Crossover in Genetic Programming. Technical Report CSRP-98-7. The University of Birmingham (UK), 1998, 21 p.
8. Machine Learning Repository UCI. Available at: <http://archive.ics.uci.edu/ml/datasets.html>.

© Христоненко Д. И., Семенкина Е. С., 2013

УДК 519.6

## SELF-CONFIGURING GENETIC PROGRAMMING ALGORITHM FOR MEDICAL DIAGNOSTIC PROBLEMS

E. A. Popov, M. E. Semenkina

Siberian State Aerospace University named after academician M. F. Reshetnev  
31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660014, Russian Federation  
E-mail: [epopov@bmail.ru](mailto:epopov@bmail.ru), [semenkina88@mail.ru](mailto:semenkina88@mail.ru)

*Genetic programming algorithm for neural network automatic design is suggested. Ensemble member competence estimations based procedure of decision making with intelligent information technologies is proposed. Effectiveness of the approach is proved on the benchmark and real-world medical diagnostic problems.*

**Keywords:** intelligent information technologies, genetic programming algorithm, ensemble based decision making.

## САМОКОНФИГУРИРУЕМЫЙ АЛГОРИТМ ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ ДЛЯ ЗАДАЧ МЕДИЦИНСКОЙ ДИАГНОСТИКИ

Е. А. Попов, М. Е. Семенкина

Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева  
Российская Федерация, 660014, Красноярск, просп. им. газ. «Красноярский рабочий», 31  
E-mail: epopov@bmail.ru, semenkina88@mail.ru

*Рассматривается алгоритм генетического программирования для автоматизированного проектирования нейронных сетей. Предлагается алгоритм генетического программирования для автоматического генерирования ансамблей интеллектуальных информационных технологий. Эффективность подхода проверена на тестовых и прикладных задачах медицинской диагностики.*

*Ключевые слова: интеллектуальные информационные технологии, алгоритм генетического программирования, коллективное принятие решений.*

Recently, the importance of information support of various medical technologies is steadily increasing. The psychological aspect of the information technology application for medical purposes has a great importance in association with the doctor work peculiarities. In this case reasons for the decision are very important, especially if it is offered by computer [1].

Although genetic programming (GP) [2] have been successfully used in solving many real world problems, the performance of this technique essentially depends on the selection of its settings and tuning parameters. The process of settings determination and parameters tuning is known to be a time-consuming and complicated task. That forces the end user to worry about the decision-making quality. This research devoted to “self-adapted” GP is based on a range of ideas aimed at reducing the human expert role in algorithm designing. This should increase the validity of the decision-making process.

The rest of the paper is organized in following way. Section 1 describes the proposed method for GP self-configuring as well as testing results over the benchmark symbolic regression problems. Neural networks design with GP is described in Section 2, in Section 3 the ensemble design method is described, Section 4 contains the performance comparison of suggested algorithms and alternative approaches on three real world medical diagnostic problem. In Conclusion we discuss obtained results and directions of the future research.

**Self-configuring genetic programming algorithm.** It is necessary to find solution for the main problem of evolutionary algorithms before suggesting for end users to apply genetic programming algorithm for automated design of tools for solving classification problems. Choosing the right GP setting for each problem is a difficult task even for experts in the field of evolutionary computation, so we need to find a way to avoid it.

In our algorithm [3] operators’ probabilistic rates dynamic adaptation on the level of population with centralized control techniques was applied (see Fig.1). Instead of the real parameters adjusting, setting variants were used, namely types of selection (fitness proportional, rank-based, and tournament-based with three tournament sizes), crossover (one-point, two-point, as well as equiprobable, fitness

proportional, rank-based, and tournament-based uniform crossovers [3]), population control and level of mutation (medium, low, high for two mutation types). Each of these has its own initial probability distribution (fig. 2) which are changed as algorithm executes (fig. 3).

The uniform crossover operator in evolutionary algorithms is known as one of the most effective crossover operators. For this reason, uniform crossover operator for GP was modified in [3] with a purpose of improving its performance. Modification gives the possibility to fulfill uniform crossover also in case when nodes have different arity because all functions’ arguments compete with each other. E. g., if one tree is “EXP – X” ( $\exp(x)$ ) and other one is “+ – Y – Z” ( $y+z$ ) and the off-spring inherits the node “+” then resulting subtree can be one of “ $y+z$ ”, “ $x+z$ ” or “ $y+x$ ”. If the off-spring inherits the node “EXP” then resulting subtrees can be “ $\exp(x)$ ”, “ $\exp(y)$ ”, “ $\exp(z)$ ”. This modification brings more flexibility to the crossover process and adds the potential for a change in the algorithms behavior.

Selective pressure during the recombination process was introduced making the probability of a parental gene being passed to the off-spring depended on parent fitness values. The off-spring can inherit every of its nodes from one of parents not only equiprobably but also with different probabilities determined by parent fitness values in one of the ways like the usual selection operators. Fitness proportional, rank-based and tournament-based uniform crossover operators were added to the conventional operator that can be called now the equiprobable uniform crossover.

As a commonly accepted benchmark for GP algorithms is still an “open issue” [4], the symbolic regression problem with 17 test functions borrowed from [5] were used in [3] for testing the modified genetic programming algorithm with new uniform crossover operators (MGP).

Results of self-configuring GP (SelfCGP) performance evaluation over 17 test problems and the performance comparison with conventional GP are presented in Table 1 below ([3]). Experiments settings are 100 individuals, 300 generations and 100 algorithm runs for each test function. The variance is given over 17 test functions. The reliability is the portion of the runs for a given algorithm that gives satisfactorily precise solution (MSE is less than 0.01). Statistical significance was estimated with ANOVA.

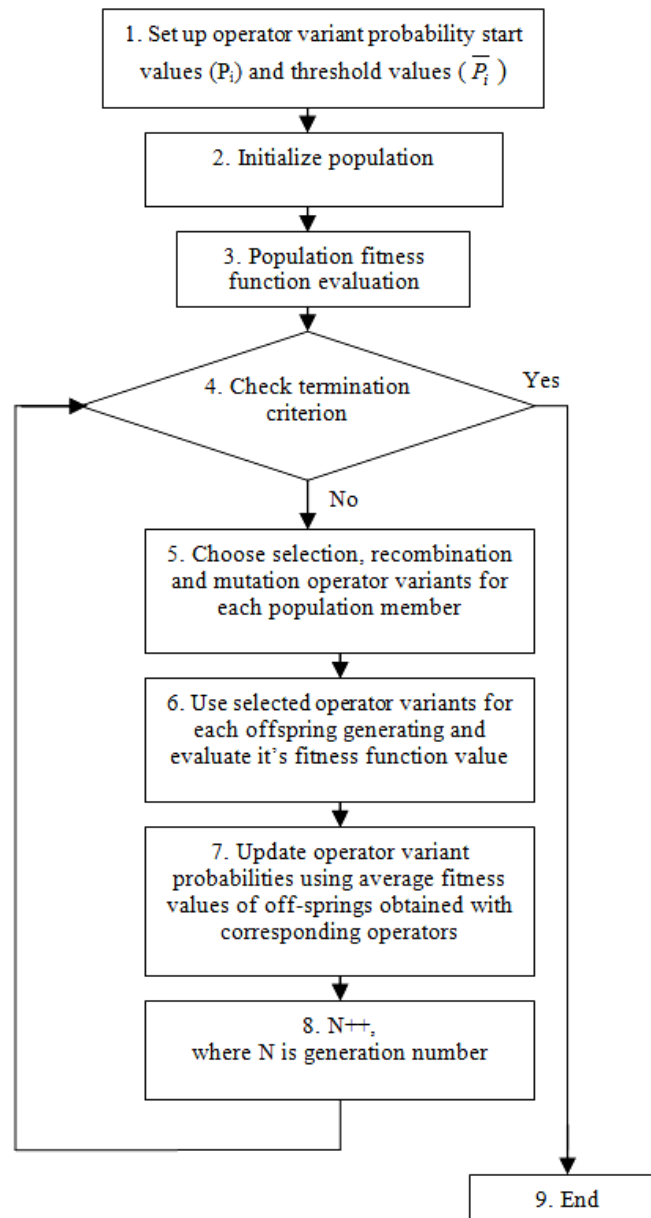


Fig. 1. Main part of SelfCGP block diagram

From table 1 we can see that GP with modified uniform crossover operators (MGP) performs better than conventional GP. This demonstrates that proposed operators may be useful.

SelfCGP reliability averaged over 17 test function is better than averaged best reliability of conventional GP and slightly less than best reliability of modified GP. The worse reliability (for the most hard problem) averaged over 100 runs is equal to 0.42. The best reliability is equal to 1.00. Computational efforts are less than alternative algorithms have. It gives us a possibility to recommend SelfCGP for solving symbolic regression problems as the better alternative to the conventional GP. Main advantage of SelfCGP is no need of algorithmic details adjustment without any losses in the performance that makes this algorithm useful for many applications where end users being no experts in evolutionary modeling nevertheless intend to apply GP for solving these problems.

Solving symbolic regression problems, it is interesting to have not only the precise enough computational procedure but also a symbolically correct answer, i. e. exactly the same analytical expression that was used to generate data base. The last three columns in table 1 contain information on the quality of the obtained approximations. The first column shows the percentage of precise solutions symbolically identical to the test function. The second one shows the percentage of conditionally precise solutions that needed some elementary transformations and numbers rounding to be symbolically identical to the test function. The third column shows the percentage of the obtained solutions which cannot be transformed into a symbolically identical form (but give enough good approximation).

We can add that the self-configuration itself (without new uniform crossover operators use) does not improve this ability of the conventional GP. Such a role of uniform

crossover operators in GP can be explained through the observation that these operators prevent GP tree's blow up which usually complicates the symbolic expression and gives no possibility to find precise solution.

**ANN design with GP.** We have to describe our way to model and optimize an ANN structure with GP before an employment of our SelfCGP algorithm.

Usually, GP algorithm works with tree representation, defined by functional and terminal sets, and exploits of the specific solution transformation operators (selection, crossover, mutation, etc.) until termination condition will be met [6].

The terminal set of our GP includes input neurons and 15 activation functions such as bipolar sigmoid, unipolar sigmoid, Gaussian, threshold function, linear function, etc. The functional set includes specific operations for neuron placement and connections. The first operation is the plac-

ing a neuron or a group of neurons in one layer. There will be no additional connections appeared in this case. The second operation is the placing a neuron or a group of neurons in sequential layers in such a way that the neuron (group of neurons) from the left branch of tree preceded by the neuron (group of neurons) from the right branch of tree. In this case, new connections will be added that connect the neurons from the left trees branch with the neurons from the right trees branch. Input neurons cannot receive any signal but have to send a signal to at least one hidden neuron. It might be so that our GP algorithm does not include some input neurons in resulting tree, i. e., high performance ANN structure that uses not all problem inputs can be found. This feature of our approach admits using our GP for the selection of the most informative problem inputs combination. Tree and corresponding neural network example are presented on the figure 4.

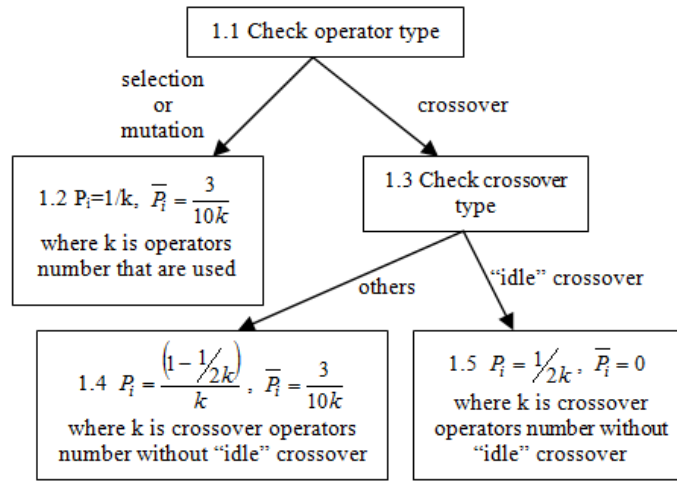


Fig. 2. Flowchart illustrating step 1 in SelfCGP block diagram

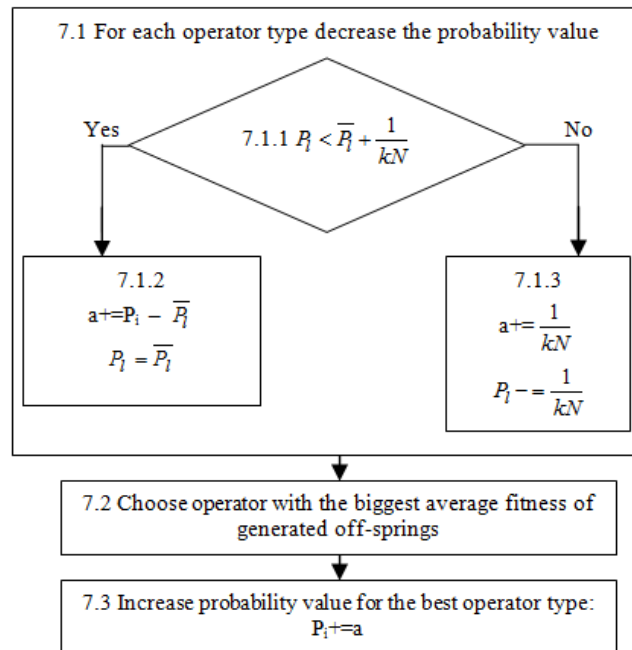


Fig. 3. Flowchart illustrating step 7 in SelfCGP block diagram

Table 1

Self-configuring GP performance over test symbolic regression problems

Algorithms	Reliability			Average generations variance	Precise solutions, %	Cond. precise solutions, %	Approx. Solutions, %
	On the most simple task	On the most difficult task	In general				
GP	0,77 [0,41; 0,91]	0,13 [0; 0,27]	0,43 [0,00; 0,91]	[33; 289]	50	16	34
MGP	0,79 [0,39; 1]	0,34 [0,15; 0,43]	0,53 [0,11; 1,00]	[27; 243]	58	20	22
SelfCGP without MGP	0,83	0,19	0,46 [0,19; 0,83]	[35; 254]	50	17	33
SelfCGP	1	0,48	0,69 [0,42; 1,00]	[49; 201]	58	16	26

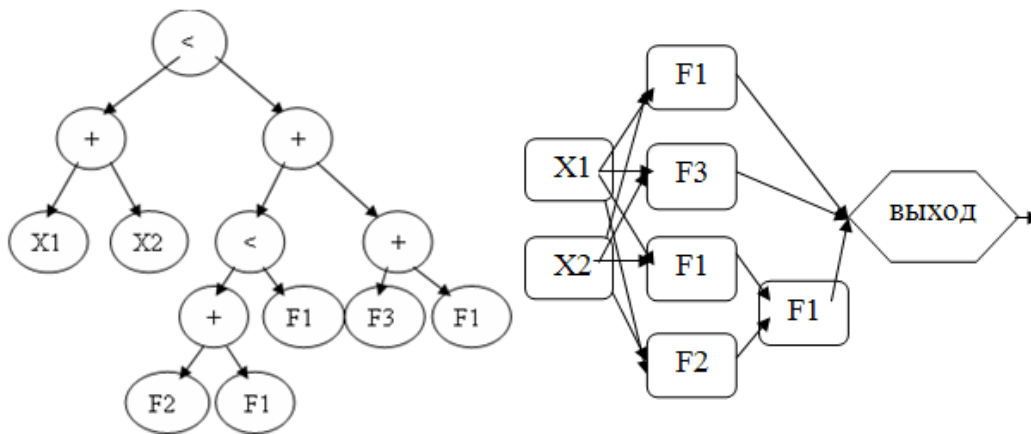


Fig. 4. Tree and corresponding neural network example

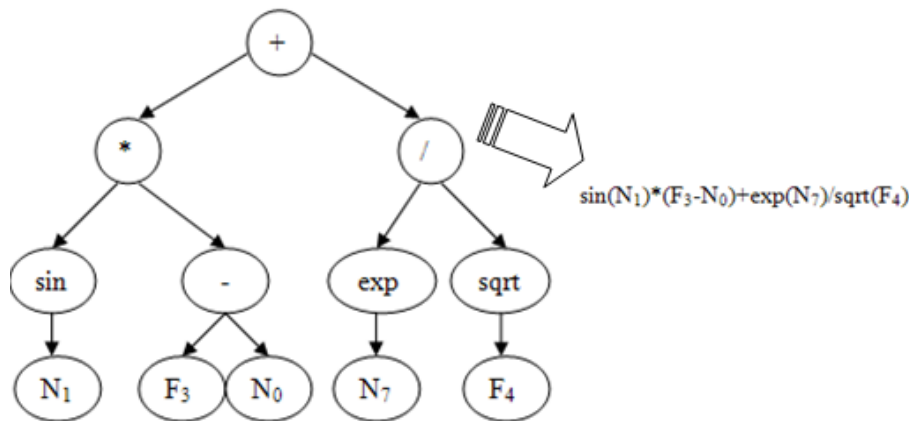


Fig. 5. Tree and corresponding decision making ensemble formula example

The GP algorithm forms the tree from which the ANN structure is derived. The ANN training is executed to evaluate its fitness that depends on its performance in solving problem in hand, e.g., approximation precision or number of misclassified instances. For training this ANN, connection weights are optimized with special self-configuring genetic algorithm (SelfCGA) that does not need any end user efforts to be the problem adjusted doing it automatically. We have no place here to go into the details of SelfCGA, referring to our other paper [7] but have to say that it is based on the same ideas as the self-configuring genetic programming algorithm described in

the pervious section. When GP finishes giving the best found ANN structure as the result, this ANN is additionally trained with again SelfCGA hybridized with local search.

The efficiency of the proposed approach was tested on a representative set of test problems (approximation, time series prediction). Averaging of the mean square error was carried out by 20 runs. The test results showed that the neural networks created by genetic programming algorithm have a small number of neurons in comparison with neural networks obtained by means of neuro-simulator and are not fully connected (few connections

between neurons). These neural networks have enough small mean square error.

**Integration of IIT ensembles with self-configuring genetic programming algorithm.** Usual practice of data analysis problems solving lies in the fact that the structure choice and parameters setting of a separate intelligent information technology (IIT) are performed and then the best found IIT solves the problem. However, the solutions quality improvement is particularly important for solving problems of medical diagnosis and it is desirable to have a set of IIT and to use all of them at the same time according to the chosen ensemble technology.

In this paper we apply self-configuring genetic programming algorithm to build a common method of decision making by IIT ensemble through the creation of symbolic regression formulas with various IIT as elements of the terminal set.

Having the developed appropriate tool for IIT automated design that does not require the effort for its adjustment, we applied our self-configuring genetic programming technique to construct the formula that shows how to compute an ensemble decision using the component IIT decisions. The algorithm involves different operations and math functions and uses the models of different kinds providing the diversity among the ensemble members. In our numerical experiments, we use symbolic expressions and neural networks, automatically designed with our SelfCGP algorithm, as the ensemble members. The algorithm automatically chooses the component IIT which are important for obtaining an efficient solution and doesn't use the others. The ensemble component IIT are taken from the preliminary IIT pool that includes 20 ANNs and 20 symbolic regression formulas (SRFs) generated in advance with SelfCGP.

Ensemble decision making method example is presented on the Fig. 5, where  $N_0, N_1, N_2, \dots, N_{10}$  are ANNs from preliminary pool,  $F_0, F_1, F_2, \dots, F_{10}$  are symbolic regression formulas from preliminary pool, at the same time better IIT has smaller number, i.e.  $N_0$  is the best ANN in preliminary pool.

The proposed procedure has been implemented as a software system and tested on a test problems representative set.

**Approbation on applied problems of medical diagnostics.** To test the proposed approach, we used three practical problems of medical diagnosis: breast cancer diagnosis (11 attributes, 2 classes, 458 patients records with benign cancer and 241 records with malignant tumors), diabetes diagnosis (9 inputs, 2 classes, 500 patients with a negative result and 268 patients positive for diabetes) and Heart Disease diagnosis (14 inputs, 2 classes, 303 instances). Input data were taken from [8]. In the problem of cancer diagnostic we need to determine the type of tumor (benign or malignant) throw the existing test result (tumor cell characteristics (shape, thickness, uniformity, etc.)). In the diabetes diagnostics problem we need to determine whether a patient is sick with diabetes (all patients are women older than 21 years) according to available data (on age, ancestry, pregnancies number, body mass index, skinfold size, the glucose amount in the plasma, the insulin amount in the blood serum, blood pressure, etc.). For these two problems evaluations were fulfilled in the same manner as in [9], i. e. indicator is equal to weighted sum of mean squared error and quality of classification. Results for comparison for the last problem were taken from [10; 11]. For this problem the indicator is the error of classification.

Table 2

Ensembling methods comparison

Classifier	Cancer	Diabetes	Heart
SelfCGP+ANNE	0	17,18	0,156
SelfCGP+ANN+SRF+Ens.	0,06	17,43	0,161
SelfCGP+SRFE	0,34	18,21	0,168
Bagging (SelfCGP+ANN)	0,67	18,22	0,171
Bagging (SelfCGP+SRF)	0,95	19,34	0,176
ANN w. a.	1,03	19,03	0,179
ANN s. a.	1,09	19,75	0,183
SRF w. a.	1,22	19,86	0,186
SRF s. a.	1,27	20,23	0,193
ANN+SRF w. a.	1,09	19,34	0,178
ANN+SRF s. a.	1,18	19,79	0,181
SelfCGP+ANN	1,05	19,69	0,185
SelfCGP+SRF	1,23	20,01	0,188
CROANN	1,06	19,67	—
GANet-best	1,06	24,70	—
COOP	1,23	19,69	—
CNNE	1,20	19,60	—
ESANN	0,95	20,93	—
GSOANN	0,65	19,79	—
NLCS (neural-based learning classifier systems)	—	—	0,16
Neural Ensemble	—	—	0,181
NeC4.5	—	—	0,194

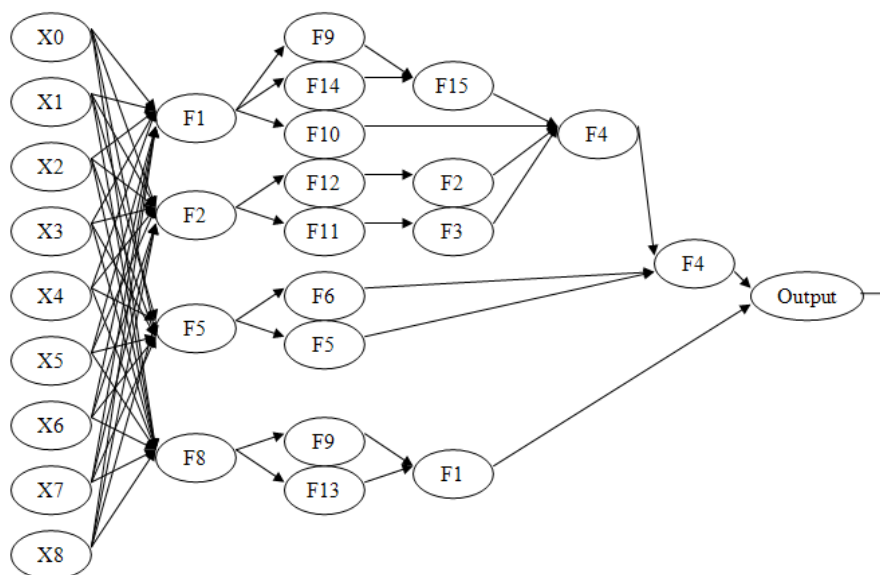


Fig. 6. Example of ANN for Diabetes diagnostic problem

For the designing every IIT, corresponding data set is randomly divided into three parts, i. e., training sample (60 %), validation sample (20 %) and test sample (20 %).

The first three lines contain results of the ensembling method suggested in this paper for three kinds of ensemble members – ANNs (ANNE), symbolic regression formulations (SRFE) and both (ANN+SRF+Ens.). Next eight lines contain results for conventional methods of ensemble forming – bagging, simple (s. a.) and weighted averaging (w. a.). Next two lines show results of single best technologies (ANN and SRF) automatically generated with SelfCGP. Results of these thirteen lines are averaged over 20 independent runs. The statistical robustness of the results obtained was confirmed by ANOVA tests which were used for processing received evaluations of our algorithms performance.

Results in table 2 demonstrate that the SelfCGP based ensembling method used the ANNs or ANNs and SRFs integration outperforms conventional ensembling methods, the single best ANN and SRF designed with SelfCGP as well as other given classification methods.

Example of ANN for Diabetes diagnostic problem is presented on the fig. 6. Ensembles obtained by solving the diabetes diagnostic problem have the form

$$\frac{F_2}{2.11} \sqrt{e^{F_8 F_7 F_4}} \quad (\text{for symbolic regression formulations})$$

$$\text{and } \sin\left(\frac{N_2 + N_8 + N_9}{1.998}\right) + 0.02 \quad (\text{for ANN}).$$

**Conclusions.** Thus developed software system for intelligent information technology ensembles automated design allows solving complex problems of medical diagnostics. IIT ensemble design increases the efficiency and reliability of IIT application. The efficiency of this approach has been verified not only on the test but also on the real practical problems that allows us to recommend it for end users as a convenient and effective tool for data mining.

The further development of the system is aimed to the expansion of its functionality by including the other types of IITs (fuzzy logic systems, decision trees, neuro-fuzzy systems, other kinds of ANNs, multiobjective selection, etc.).

## References

1. Jarikov O. G., Litvinin A. A., Kovalev V. A. Expert systems in medicine. *Medezinskie novosti*. 2008, № 10, p. 15–18 (in Russian).
2. Koza J. R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, MA, Cambridge (1992).
3. Semenkin E., Semenkin M. Self-Configuring Genetic Programming Algorithm with Modified Uniform Crossover Operator. In: Proceedings of the IEEE Congress on Evolutionary Computation (IEEE CEC), p. 1918–1923. Brisbane, Australia (2012).
4. O'Neill M., Vanneschi L., Gustafson S., Banzhaf W. Open issues in genetic programming. In: Genetic Programming and Evolvable Machines 11, p. 339–363 (2010).
5. Finck S. et al. Real-parameter black-box optimization benchmarking 2009. In: Presentation of the noiseless functions. Technical Report Research Center PPE (2009).
6. Poli R., Langdon W. B., McPhee N. F. A Field Guide to Genetic Programming. Published via <http://lulu.com> and freely available at <http://www.gp-field-guide.org.uk>, 2008 (With contributions by J. R. Koza).
7. Semenkin E. S., Semenkin M. E. Self-configuring Genetic Algorithm with Modified Uniform Crossover Operator. Y. Tan, Y. Shi, and Z. Ji (Eds.): Advances in Swarm Intelligence. Lecture Notes in Computer Science 7331. Springer-Verlag, Berlin Heidelberg, 2012, p. 414–421.
8. Frank A., Asuncion A. UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA:



University of California, School of Information and Computer Science. 2010.

9. Yu J. J. Q., Lam A. Y. S., Li V. O. K. Evolutionary Artificial Neural Network Based on Chemical Reaction Optimization. In 2011 IEEE Congress on Evolutionary Computation (CEC'2011), New Orleans, LA, 2011.

10. Dam H. H., Abbass H. A., Lokan C., Yao X. Neural-Based Learning Classifier Systems. IEEE Transactions

on Knowledge and Data Engineering. 2008, vol. 20, № 1, p. 1–14.

11. Zhi-Hua Zhou and Yuan Jiang. NeC4.5: Neural Ensemble Based C4.5. IEEE Trans. Knowl. Data Eng. 2004, № 16.

© Попов Е. А., Семенкина М. Е., 2013

УДК 005; 519.7; 303.732

## ABOUT PERFORMANCE IMPROVEMENT OF EVOLUTIONARY STRATEGIES TECHNIQUE APPLIED TO OPTIMAL CONTROL PROBLEM INDIRECT METHOD

I. S. Ryzhikov

Siberian State Aerospace University named after academician M. F. Reshetnev  
31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660014, Russian Federation.  
E-mail: ryzhikov-88@yandex.ru

*The optimal control problem for nonlinear dynamic systems is considered. The proposed approach is based on both partially analytical and partially numerical techniques of the optimal control problem solving. Optimal control problem is reduced to unconstrained extremum problem, which is related to seeking for the initial point of the co-state variables that would satisfy the boundaries. To solve the optimization problem, well-known global optimization techniques are suggested and compared. The performance of the evolutionary strategies algorithm was increased by implementing the special restarting condition in the scheme.*

*Keywords: stochastic optimization, optimal control, indirect method, variation problem, dynamic system.*

## О ПОВЫШЕНИИ ЭФФЕКТИВНОСТИ АЛГОРИТМА ЭВОЛЮЦИОННЫХ СТРАТЕГИЙ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ НЕПРЯМЫМ МЕТОДОМ

И. С. Рыжиков

Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева  
Российская Федерация, 660014, Красноярск, просп. им. газ. «Красноярский рабочий», 31  
E-mail: ryzhikov-88@yandex.ru

*Рассматривается задача нахождения оптимального управления для нелинейных динамических систем. Предложенный подход основан на частично аналитическом и частично численном решении задачи оптимального управления. Исходная задача сводится к задаче поиска экстремума функций без ограничений, решением которой являются начальные координаты для сопряженных переменных, при которых удовлетворяются граничные условия. Для решения приведенной задачи сравнивались различные широко известные методы глобальной оптимизации. Эффективность метода эволюционных стратегий была повышена через введение специального условия на перезапуск алгоритма.*

*Ключевые слова: стохастическая оптимизация, оптимальное управление, косвенный метод, вариационная задача, динамическая система.*

In this paper the idea of modified evolutionary strategies algorithm performance improvement is investigated. Some hypotheses about the ways to increase the efficiency were put forward. In previous work the different evolutionary and nature-based algorithm were examined and it was shown that these techniques are not reliable, so there is a need in special operands to be implemented. Current work consists of problem definition, previous results and suggested improvements.

The optimal control problem for dynamic systems

with one control input and integral functional is considered. Since the problem is old and it originates from the practical needs, there exist many techniques to solve the optimal control problem in different problem definitions and for different systems. But the developing of the modern technologies creates new optimal control problems that cannot be solved via well-known and classical approaches. The main problem is nonlinearity of the system model or the criterion. In general case, there is no universal analytical technique that guarantees the solution of