

УДК 517.9

Cooperation of Bio-inspired and Evolutionary Algorithms for Neural Network Design

Shakhnaz A. Akhmedova*

Vladimir V. Stanovov[†]

Eugene S. Semenkin[‡]

Reshetnev Siberian State University of Science and Technology
Krasnoyarskiy Rabochiy, 31, Krasnoyarsk, 660037
Russia

Received 30.06.2017, received in revised form 12.09.2017, accepted 20.01.2018

A meta-heuristic called Co-Operation of Biology-Related Algorithms (COBRA) with a fuzzy controller, as well as a new algorithm based on the cooperation of Differential Evolution and Particle Swarm Optimization (DE+PSO) and developed for solving real-valued optimization problems, were applied to the design of artificial neural networks. The usefulness and workability of both meta-heuristic approaches were demonstrated on various benchmarks. The neural network's weight coefficients represented as a string of real-valued variables are adjusted with the fuzzy controlled COBRA or with DE+PSO. Two classification problems (image and speech recognition problems) were solved with these approaches. Experiments showed that both cooperative optimization techniques demonstrate high performance and reliability in spite of the complexity of the solved optimization problems. The workability and usefulness of the proposed meta-heuristic optimization algorithms are confirmed.

Keywords: co-operation, bio-inspired algorithms, differential evolution, neural networks, classification.

DOI: 10.17516/1997-1397-2018-11-2-148-158.

Introduction

Modern zero-order optimization techniques often introduce cooperation between several well-known optimizers to use the advantages of each method. Self-configuring and co-evolutional [1] genetic algorithms are good examples of such successful approaches, as well as the cooperation of biology-related algorithms [2]. The main reason why these techniques work well is the fact that each separate set of genetic operators or each biology-inspired method has its own unique characteristics, which are useful during different stages of the optimization process. If the cooperation scheme is tuned well, so that it is capable of tracking the success rate of each algorithm and adjusting their resources, then the cooperation usually shows better results than each individual algorithm on average.

If we consider a known algorithm, the Co-operation of Biology-Related Algorithms, COBRA [2], we may see that its components are very similar in their structure, i.e. all these algorithms are PSO-like optimization techniques (here "PSO" refers to the Particle Swarm Optimization algorithm [3]). Similarly, co-evolutionary genetic algorithms, as well as self-configuring genetic

*shahnaz@inbox.ru

[†]vladimirstanovov@yandex.ru

[‡]eugeneseamenkin@yandex.ru

© Siberian Federal University. All rights reserved

algorithms, use different variations of the same optimization method. However, there is a possibility to use component algorithms which have a different structure. For certain reasons, such co-operative optimization algorithms are not often developed, most likely due to the fact that implementing such methods requires knowledge from different spheres.

In this paper we introduce the cooperation of two optimization methods, namely Differential Evolution [4] and Particle Swarm Optimization [3] (which will be referred to as "DE+PSO"), with a cooperation scheme similar to one used in the COBRA algorithm. We demonstrate that this approach is capable of obtaining results comparable to those achieved with other optimization methods. Moreover, in this paper the workability and usefulness of the developed meta-heuristics are demonstrated not only on benchmark problems but also on much harder optimization problems related to designing the structure of artificial neural network (ANN) based classifiers and adjusting the weight coefficients.

Thus, the rest of the paper is organized as follows. Firstly, a description of the proposed optimization techniques (COBRA and DE+PSO) is presented. Then the procedure of the neural network design is explained. Following this, experimental results are demonstrated, whereby the workability of the meta-heuristics is demonstrated with the ANN-based classifier design for two real-world classification problems. The conclusion contains a discussion of results and considers further research directions.

1. Co-Operation of Biology-Related Algorithms

1.1. Original version

The meta-heuristic approach called Co-Operation of Biology-Related Algorithms or COBRA [2] was developed based on five optimization methods, namely Particle Swarm Optimization (PSO) [3], the Wolf Pack Search (WPS) [5], the Firefly Algorithm (FFA) [6], the Cuckoo Search Algorithm (CSA) [7] and the Bat Algorithm (BA) [8] (hereinafter referred to as "component-algorithms"). Later the Fish School Search (FSS) [9] was also added as one of COBRA's component-algorithms. The main reason for the development of a cooperative meta-heuristic was the inability to say which of the above-listed algorithms is the best one or which algorithm should be used for solving any given optimization problem [2]. Thus, the idea was to use the cooperation of these bio-inspired algorithms instead of any attempts to understand which one is the best for the problem in hand.

The originally proposed approach consists in generating one population for each bio-inspired algorithm, therefore six populations, which are then executed in parallel, cooperating with each other. The COBRA algorithm is a self-tuning meta-heuristic, so there is no need to choose the population size for each component-algorithm. The number of individuals in the population of each algorithm can increase or decrease depending on the fitness values: if the overall fitness value was not improved during a given number of iterations, then the size of each population increased, and vice versa.

There is also one more rule for population size adjustment, whereby a population can "grow" by accepting individuals removed from other populations. The population "grows" only if its average fitness value is better than the average fitness value of all other populations. Therefore, the "winner algorithm" can be determined as an algorithm whose population has the best average fitness value. The described competition among component-algorithms allows the biggest population size to be allocated to the most appropriate bio-inspired algorithm on the current

generation.

The main goal of communication between all populations is to prevent their preliminary convergence to their own local optimum. Such "communication" was determined in the following way: populations exchange individuals, whereby a part of the worst individuals of each population is replaced by the best individuals of other populations. Thus, the group performance of all algorithms can be improved.

The performance of the proposed algorithm was evaluated on a set of unconstrained real-parameter optimization problems. Experiments showed that COBRA works successfully and is reliable on these benchmarks. Results also showed that COBRA outperforms its component algorithms when the dimension grows and more complicated problems are solved.

1.2. Fuzzy controller

The main idea of using a fuzzy controller is to implement a more flexible tuning method, compared to the original COBRA tuning algorithm. Fuzzy controllers are widely known for their ability to generate real-valued outputs using special fuzzification, inference and defuzzification schemes [10]. In this work, success rates were used as inputs and population size changes as outputs.

The fuzzy controller had 7 input variables, including 6 success rates of each component and an overall success rate, and 6 output variables, i.e. the number of solutions to be added to or removed from each component. The components' success rates (thus the first 6 input variables) were determined as the best fitness value of the corresponding population. Finally, the last input variable was determined as the ratio of the number of iterations, during which the best found fitness value was improved, to the given number of iterations, which was a constant period. Thus, the process of the population growth was automated by the fuzzy controller.

The Mamdani-type fuzzy inference was used to determine the value of outputs, and the rules had the following form:

$$R_q : IF x_1 \text{ is } A_{q1} \text{ and } \dots \text{ and } x_n \text{ is } A_{qn} \text{ THEN } y_1 \text{ is } B_{q1} \text{ and } \dots \text{ and } y_k \text{ is } B_{qk}, \quad (1)$$

where R_q is the q -th fuzzy rule, $x = (x_1, \dots, x_n)$ is the set of input values in n -dimensional space ($n = 7$ in this case), $y = (y_1, \dots, y_k)$ is the set of outputs ($k = 6$), A_{qi} is the fuzzy set for the i -th input variable, B_{qj} is the fuzzy set for the j -th output variable.

The rule base contained 21 fuzzy rules, which had the following structure: each 3 rules described the case when one of the components gave better results than the others (as there were 6 components, 18 rules were established); the last 3 rules used the overall success of all components (variable 7) to add or remove solutions from all components, i.e. to regulate the computational resources.

The input variables were always in the range $[0, 1]$, and fixed fuzzy terms of triangular shape were used for this case. In addition to the three classical fuzzy sets A_1 , A_2 and A_3 , the "Don't Care" (DC) condition and the A_4 term with the meaning "larger than 0" (opposite to A_1) were also used to decrease the number of rules and make them simpler.

For the outputs, 3 fuzzy terms of triangular shape were used. The output fuzzy terms were symmetrical, and the positions and shapes were determined by two values, encoding the left and right position of the central term, as well as the middle position of the side terms in one value, and the left and right positions of the side term in another value. These two values were optimized using the PSO algorithm. The defuzzification procedure was performed by calculating the centre of mass of the shape received by fuzzy inference.

2. Cooperation of differential evolution and particle swarm optimization

In this paper, a meta-heuristic approach based on the cooperation of two component algorithms, namely Differential Evolution (DE) [4] and Particle Swarm Optimization (PSO) [3], denoted as DE+PSO, was proposed. These two algorithms were chosen as they are well-known, work well for real-parameter optimization, and represent techniques based on different ideas. A large variety of cooperative optimization methods developed previously use algorithms of similar structure.

The component algorithms were used in their basic variant, i.e. described in the original papers. The parameters used for DE were chosen as recommended: $F = 0.5$, $Cr = 0.5$. For PSO, the following set of parameters was used: $c_1 = c_2 = 2.05$, $w = 0.7298$. The DE+PSO algorithm consisted of the following steps:

1. Initialize PSO population by randomly setting the coordinates and velocity of the particles.
2. Initialize DE population by randomly setting individual coordinates.
3. Evaluate both populations, find the best solutions in each.
4. Determine the winner-algorithm and save the best solution separately.
5. If the stopping criteria are satisfied, finish calculations and return the best solution.
6. Evaluate algorithms' success rates depending on previous best found solutions.
7. Change the population sizes depending on which algorithm is the winner, and the success rates of each component.
8. Generate new particles/individuals by using random initialization, or remove the worst particles/individuals.
9. Perform generation of new particle coordinates and velocities by applying the standard PSO scheme, and evaluate the particles.
10. Perform generation of new individuals of the DE algorithm using the standard DE scheme, and evaluate the particles.
11. Go to step 5.

The population size adjustment is performed depending on the effectiveness of each component. The effectiveness is measured using the best fitness values achieved over the last 7 iterations. In the case of there having been no improvement during last 7 iterations, the size of the population of the algorithm is increased, because we consider that the optimization problem is too complicated to be solved with a small number of points. In the case of there not having been at least one improvement of fitness value over the last 7 iterations, the population size is decreased. The population size is increased by 2 and decreased by 1.

The concurrence between the component algorithms is maintained by means of determining the winner algorithm on all iterations. The winner algorithm is defined as the algorithm having the best fitness value at the current generation. The winner algorithm gets its population size increased, while the other algorithm loses the same number of points as was gained by the winner. If both algorithms have the same fitness, then neither of them is the winner. The number of points gained and lost is always equal to 2.

The migration between the components is performed by the exchange of best solutions received by component algorithms every 10 generations. The starting population size for both DE and PSO is equal to 75, the minimal size is 25 and the maximal size is 300. The experimental results presented in the corresponding section show that the winner algorithm changes depending on the function optimized.

3. Artificial neural network design

Artificial neural network (ANN) models are frequently used for solving various data mining problems such as classification and prediction among others. In this study, only classification problems solved by two types of ANN models are considered.

ANN-based classifiers have three primary components: the input data layer, the hidden layer(s) and the output layer. Each of these layers contains nodes and these nodes are connected to nodes at adjacent layers. In addition, each node has its own activation function. Therefore, the number of hidden layers, the number of nodes, which are also called neurons, and the type of activation function on each node will be denoted as "ANN structure".

The nodes in the network are interconnected and each connection has a weight coefficient; the number of these coefficients depends on the problem being solved (number of inputs) and the number of hidden layers and neurons. Therefore, networks with a relatively complex structure usually have many weight coefficients that should be adjusted.

Thus, the neural network structure design and tuning of weight coefficients are considered as the solving of two unconstrained optimization problems: the first one with binary variables and the second one with real-valued variables. The type of variables depends on the representation of the ANN structure and coefficients.

As was stated earlier, two types of ANN models are used in this work: the first one is designed by the COBRA approach and the second is obtained by the DE+PSO algorithm.

For the ANN-based classifiers designed by the COBRA approach the maximum number of hidden layers is equal to 5 and the maximum number of neurons on each hidden layer is also equal to 5, so the overall maximum number of neurons is equal to 25. Each node is represented by a binary string of length 4. If the string consists of zeros ("0000") then this node does not exist on the respective layer. In this way, the whole structure of neural network is represented by a binary string of length 100 (25×4), and each 20 variables represent one hidden layer. The number of input neurons depends on the problem in hand, ANNs have one output neuron. Besides, 15 well-known activation functions are used for nodes: the sigmoidal function, the linear function, the hyperbolic tangent function and others (the whole list of used activation functions is given in [14]). For determining which activation function will be used on a given node the integer that corresponds to its binary string is calculated. E.g., if a neuron has the binary string "0110", then the integer is calculated in the following way:

$$0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 = 6. \quad (2)$$

Therefore for this neuron we use the sixth activation function from the given list.

Thus, the optimization method for unconstrained problems with binary variables COBRA-b [11] is implemented for the best ANN structure selection and the fuzzy controlled COBRA is applied for every structure weight coefficient adjustment.

For the second type of ANN models, the following assumptions are made. The neural network is fully-connected, contains only one hidden layer with 15 neurons, and does not use biases. All neurons in the hidden layer use a logistic function as activation. As an output layer, the softmax layer is used, which performs calculation of probabilities for all classes (2 in our case) using the following equation:

$$P(y = i|x) = \frac{\exp(x^T \times w_i)}{\sum_{j=1}^K \exp(x^T \times w_j)}, \quad (3)$$

where j is the class number, y is the output of the network, x is the input to the softmax layer, and w is the weight matrix. The cross-entropy loss function value is used as an estimation of prediction quality and is calculated by using the following equation:

$$H(p, q) = - \sum_{i=1}^K p_i \times \log(q_i), \quad (4)$$

where p_i is the probability calculated by the softmax layer, and q_i is the value from the one-hot encoding of class numbers. This encoding represents a $N \times K$ matrix, in which N is the number of instances and K is the number of classes, where each row contains 1 at the corresponding class number, while all the other values in this row are zeros.

Using softmax layer with cross-entropy as a loss function is preferable due to the fact that this function is smooth, unlike the usual classification error rate, which has plateaus and steps. Thus, the second type of ANN-based classifiers has an established structure and the weight coefficients are adjusted by the DE+PSO meta-heuristic approach.

4. Experimental results

4.1. Optimization problems

In this study, the following 10 benchmark problems taken from [12] were used in experiments: the Rotated Discus Function, the Different Powers Function, the Rotated Rosenbrock's Function, Schwefel's Function, the Rotated Ackley's Function, the Rotated Griewank's Function, the Rotated Katsuura Function, the Rotated Lunacek bi-Rastrigin Function, the Rotated Weierstrass Function and Rastrigin's Function. These benchmark functions were considered to evaluate the robustness of the fuzzy controlled COBRA and the DE+PSO approach. Both algorithms were tested on the above-listed benchmark functions with $D = 10$ variables. There were 51 program runs for each optimization problem and calculations were stopped if the number of function evaluations was equal to $10000D$.

The changes in population sizes are similar to those of the COBRA approach, with the only difference being that DE+PSO has fewer component algorithms. However, we may observe that on some functions the DE component appears to perform better (Schwefel's Function 2a), 2b)), while for others the PSO shows better results. At the same time, Fig. 2a) shows that even if one of the components has "won", it is not certain that it will stay the winner until the end of computation. We may observe that the size of the PSO population increases whereas the size of the DE population stays the same or decreases.

To compare results of both DE+PSO and COBRA when optimizing real-valued functions we performed tests on 10-dimensional and 30-dimensional space. In Tab. 1, the results obtained by the fuzzy controlled COBRA and DE+PSO meta-heuristic approach for $10D$ are presented. The following notations are used: the best found function value (Best), the function value averaged by the number of program runs (Mean) and the standard deviation (STD).

Fig. 1 shows the change of the COBRA component population sizes during the optimization process on two functions, Schwefel's Function (1a), 1c), 1e)) and the Different Powers Function (1b), 1d), 1f)).

Fig. 2 shows the change in the component population sizes of the DE+PSO approach during the optimization process on two functions, Schwefel's Function (2a), 2b)) and the Different Powers Function (2c)).

In Tab. 1, the winner algorithm is marked in bold for each test function, for both the best found values and mean values. The DE+PSO is the winner for 3 functions out of 10 if we are

considering mean values, and for 5 functions out of 10 if we are considering the best values found over all runs of the algorithms. The situation does not change very much if we switch to 30-dimensional space (Tab. 2).

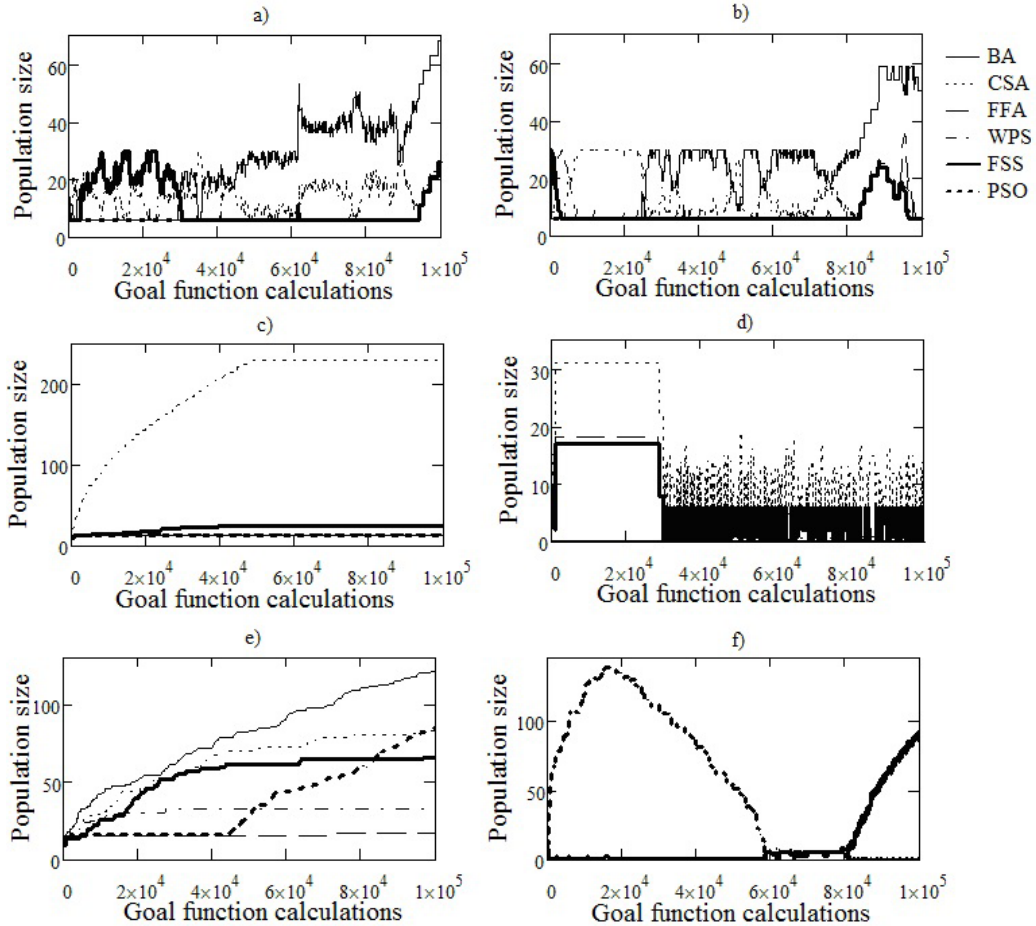


Fig. 1. Graphs of population size change for the COBRA approach

As before, bold letters show better results for a particular algorithm. DE+PSO is better for 5 functions out of 10 if we are considering mean values, and for 4 functions out of 10 if we are considering the best values. The main reason for this behaviour is probably the fact that DE+PSO has only 2 component algorithms, while COBRA has 6 of them and moreover, uses an optimized fuzzy controller that changes the sizes of all component populations.

4.2. Classification problems

In order to load the developed optimization techniques with a very hard task, two classification problems were chosen: the MAGIC Gamma Telescope problem and the Phoneme problem [13]. This choice was influenced by the fact that these problems had been solved by other researchers many times with different methods. Thus, there are many results obtained by alternative approaches that can be used for comparison.

Firstly, the Phoneme problem was solved with ANN-based classifiers. This database was used in the European ESPRIT 5516 project: ROARS [13]. The aim of this project was the

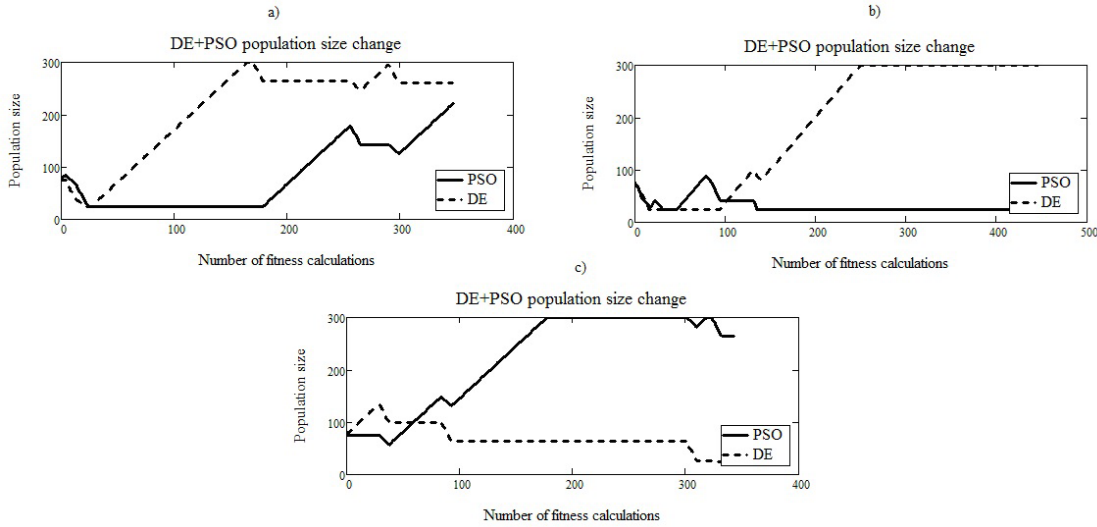


Fig. 2. Graphs of population size change for the DE+PSO approach

 Table 1. Results obtained by the fuzzy controlled COBRA with average values for successfulness evaluation, $10D$

<i>Func</i>	<i>COBRA</i>			<i>DE+PSO</i>		
	<i>Best</i>	<i>Mean</i>	<i>STD</i>	<i>Best</i>	<i>Mean</i>	<i>STD</i>
1	0.00148	0.03319	0.11152	696.966	7938.187	6822.876
2	0.00138	0.00158	0.00014	0	0	0
3	6.733e-6	2.093e-5	8.304e-6	0.01575	1.02124	1.573946
4	9.49093	101.998	241.5	15.3462	250.1268	210.6275
5	20.0832	20.1078	0.02436	20.18	20.38666	0.08296
6	7.46711	17.2175	9.3068	0.10097	0.784695	0.59639
7	0.4889	0.48966	0.00083	0.359634	1.03128	0.318048
8	10.0177	10.0948	0.4421	408.931	426.7397	8.804002
9	3.62965	4.36357	0.60906	1.86642	5.144066	1.779369
10	4.61672	6.34905	1.86941	0	2.243501	1.533782

development and implementation of a real-time analytical system for French and Spanish speech recognition. The aim of the Phoneme database is to distinguish between nasal and oral vowels. This database contains vowels coming from 1809 isolated syllables. Five different numerical attributes were chosen to characterize each vowel: they are the amplitudes of the five first harmonics, normalized by the total energy. Each harmonic is signed: positive when it corresponds to a local maximum of the spectrum and negative otherwise. Three observation moments have been kept for each vowel to obtain 5427 different instances. From these 5427 initial values, 23 instances for which the amplitude of the 5 first harmonics was zero were removed, leading to 5404 instances in the present database.

Then the MAGIC Gamma Telescope (MGT) problem was solved. The data were generated to simulate registration of high-energy gamma particles in a ground-based atmospheric Cherenkov gamma telescope using the imaging technique [13]. The Cherenkov gamma telescope observed high energy gamma rays, taking advantage of the radiation emitted by charged particles

Table 2. Results obtained by the fuzzy controlled COBRA with average values for successfulness evaluation, 30D

<i>Func</i>	<i>COBRA</i>			<i>DE+PSO</i>		
	<i>Best</i>	<i>Mean</i>	<i>STD</i>	<i>Best</i>	<i>Mean</i>	<i>STD</i>
1	0.01533	0.37206	1.65313	25257.1	64229.88	27481.72
2	0.01581	0.01768	0.00158	0	0	0
3	0.00024	0.00059	4.421e-5	8.51256	26.63419	17.60181
4	275.284	651.786	114.227	394.539	1181.418	456.6138
5	20.8419	21.0862	0.10058	20.812	20.95772	0.05761
6	9.47809	27.0826	44.0647	0.027106	0.125361	0.070245
7	1.08536	1.10279	0.01817	0.753047	2.191093	0.557353
8	30.6449	33.0245	7.30007	482.158	551.6246	38.90682
9	35.4404	36.8419	0.76274	22.5815	32.4709	4.694926
10	5.86151	10.2876	3.67494	6.96476	31.63147	13.15674

produced inside the electromagnetic showers initiated by the gammas and developing in the atmosphere. For this database, 19020 instances were obtained. Each instance was characterized by 10 numerical attributes. Moreover, there were also two classes: "gamma (signal)" and "hadron (background)".

Both datasets were taken from [13]. From the point of view of optimization, neural networks obtained by the COBRA approach had 130 and 155 real-valued variables for weight coefficients while solving the Phoneme and the MAGIC Gamma Telescope problems respectively. Besides, for the structure selection there were 100 binary variables regardless of the classification problem. For the final weight coefficient adjustment (for the best-obtained structure), the maximum number of function evaluations was set equal to 10000. The total number of connections to be tuned by the optimization procedure for the neural networks obtained by the DE+PSO approach was the following: for the Phoneme dataset – 122, and for the Magic dataset – 197.

The obtained results are demonstrated in Tab. 3 where the proportion of correctly classified instances from testing sets (%) is presented.

Table 3. Results obtained by the fuzzy controlled COBRA with average values for successfulness evaluation, 30D

<i>Classifier</i>	<i>Phoneme</i>		<i>MGT</i>	
	<i>Best</i>	<i>Mean</i>	<i>Best</i>	<i>Mean</i>
ANN+COBRA	81.68	79.14	83.88	83.18
ANN+(DE+PSO)	83.63	80.88	84.91	84.04
HEFCA [14]	83.17	80.81	84.06	82.72
SVM	85.55	83.38	67.56	65.89
Logistic regression	77.96	74.96	80.33	78.92

For comparison with other approaches, we have taken two classical methods, namely SVM and logistic regression, as well as the fuzzy classification method, HEFCA, presented in [14]. The best algorithm in every column is marked in bold. SVM appears to be the best algorithm for the Phoneme problem, however for the Magic problem it shows the worst results. The new approach using the DE+PSO algorithm for ANN training obtained the best accuracy for the Magic problem, and the second-best accuracy for the Phoneme dataset. ANN+COBRA is worse

than ANN+(DE+PSO), probably because DE is more efficient for weight tuning due to the fact that it estimates the gradient using individuals from the population, and PSO has the speed for each particle, which is similar to the momentum method used for classical ANN training. Using these properties of the DE+PSO algorithm, it is possible to train ANN more efficient than ANN+COBRA, because COBRA only has PSO-like components.

Conclusion

In this paper, a new meta-heuristic, denoted as DE+PSO, which was developed based on the cooperation idea firstly introduced in [2] for the COBRA algorithm, was described. A brief description of the fuzzy controlled COBRA was also given. The performance estimation of the proposed algorithms on sets of test functions was illustrated.

Then the described optimization methods were used for the automated design of two types of ANN-based classifiers. A binary modification of COBRA was used for the structure optimization of the first type of classifiers and the fuzzy controlled COBRA was used for the weight coefficient adjustment both within the structure selection process and for the final tuning of the best selected structure. The second type of classifiers used a softmax layer as an output layer and weight coefficients were tuned by the DE+PSO approach. These classification techniques were applied to two problems: the Phoneme and the MAGIC Gamma Telescope.

The solving of these problems is equivalent to solving big and hard optimization problems where objective functions have many (up to 197) variables and are given in the form of a computational program. The proposed algorithms successfully solved all the problems designing classifiers with competitive performance that allows the study results to be considered as confirmation of the reliability, workability and usefulness of the algorithms in solving real-world optimization problems.

Research is performed with the support of the Ministry of Education and Science of Russian Federation within State Assignment project no. 2.1680.2017/II Ч.

References

- [1] M.A.Potter, K.A.DeJong, A Cooperative Coevolutionary Approach to Function Optimization, Parallel Problem Solving from Nature – PPSN III, 1994, 249–257.
- [2] Sh.Akhmedova, E.Semenkin, Co-Operation of Biology Related Algorithms, Proceedings of the IEEE Congress on Evolutionary Computation, 2013, 2207–2214.
- [3] J.Kennedy, R.Eberhart, Particle Swarm Optimization, Proceedings of the IV International Conference on Neural Networks, 1995, 1942-1948.
- [4] R.Storn, K.Price, Differential evolution — a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, **11**(1997), no. 4, 341–359.
- [5] Ch.Yang, X.Tu, J.Chen, Algorithm of Marriage in Honey Bees Optimization Based on the Wolf Pack Search, Proceedings of the International Conference on Intelligent Pervasive Computing, 2007, 462–467.
- [6] X.S.Yang, Firefly algorithms for multimodal optimization, Proceedings of the 5th Symposium on Stochastic Algorithms, Foundations and Applications, 2009, 169–178.

- [7] X.S.Yang, S.Deб, Cuckoo Search via Levy flights, Proceedings of the World Congress on Nature and Biologically Inspired Computing, 2009, 210–214.
- [8] X.S.Yang, A new metaheuristic bat-inspired algorithm, *Nature Inspired Cooperative Strategies for Optimization, Studies in Computational Intelligence*, **284**(2010), 65–74.
- [9] F.C.Bastos, N.F.Lima, Fish School Search: an overview, *Nature-Inspired Algorithms for Optimization. Series: Studies in Computational Intelligence*, **193**(2009), 261–277.
- [10] C.-C.Lee, Fuzzy logic in control systems: fuzzy logic controller – parts 1 and 2, *Transactions on Systems, Man, and Cybernetics*, **20**(1990), no. 2, 404–435.
- [11] Sh.Akhmedova, E.Semenkin, Co-Operation of Biology Related Algorithms Meta-Heuristic in ANN-Based Classifiers Design, Proceedings of the World Congress on Computational Intelligence, 2014, 867–873.
- [12] J.J.Liang, B.Y.Qu, P.N.Suganthan, A.G.Hernandez-Diaz, Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization. Technical Report, Zhengzhou University, Nanyang Technological University, Singapore, 2012, 867–873.
- [13] A.Frank, A.Asuncion, UCI Machine Learning Repository, Available at (accessed 2010): <http://archive.ics.uci.edu/ml>.
- [14] E.Semenkin, V.Stanovov, Fuzzy Rule Bases Automated Design with Self-Configuring Evolutionary Algorithm, Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics, 2014, 318–323.

Кооперация бионического и эволюционного алгоритмов для задач проектирования искусственных нейронных сетей

Шахназ А. Ахмедова
Владимир В. Становов
Евгений С. Семенкин

Сибирский государственный университет науки и технологий им. М. Ф. Решетнева
Красноярский рабочий, 31, Красноярск, 660037
Россия

Разработанные кооперативный бионический алгоритм (COBRA) на основе нечеткого контроллера и новый коллективный алгоритм на базе дифференциальной эволюции и метода роя частиц (DE+PSO) для решения задач оптимизации функций вещественных переменных были применены для проектирования искусственных нейронных сетей. Работоспособность и целесообразность применения обеих мета-эвристик были продемонстрированы на множестве тестовых задач. Весовые коэффициенты нейронных сетей были представлены в виде вещественных переменных, которые настраивались алгоритмами COBRA с нечетким контроллером или DE+PSO. Полученными нейросетями были решены две задачи классификации (задачи распознавания образов и речи). Исследования показали, что оба алгоритма работают эффективно, несмотря на сложность задач. Таким образом, была подтверждена их работоспособность на практических задачах.

Ключевые слова: кооперация, бионические алгоритмы, дифференциальная эволюция, нейронные сети, классификация.