2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Co-Operation of Biology Related Algorithms Meta-Heuristic in ANN-Based Classifiers Design*

Shakhnaz Akhmedova
System Analysis and Operation Research Department
Siberian State Aerospace University
Krasnoyarsk, Russia
shahnaz@inbox.ru

Eugene Semenkin
System Analysis and Operation Research Department
Siberian State Aerospace University
Krasnoyarsk, Russia
eugenesemenkin@yandex.ru

***Abstract*—Meta-heuristic called Co-Operation of Biology Related Algorithms (COBRA), that has earlier demonstrated its usefulness on CEC'2013 real-valued optimization competition benchmark, is applied to ANN-based classifiers design. The basic idea consists in representation of ANN's structure as a binary string and the use of the binary modification of COBRA for the ANN's structure selection. Neural network's weight coefficients represented as a string of real-valued variables are adjusted with the original version of COBRA. Four benchmark classification problems (two bank scoring problems and two medical diagnostic problems) are solved with this approach. Multilayered feed-forward ANNs with maximum 5 hidden layers and maximum 5 neurons on each layer are used. It means that ANN's structure optimal selection requires solving an optimization problem with 100 binary variables. Fitness function calculation for each bit string requires solving an optimization problem with up to 225 real-valued variables. Experiments showed that both variants of COBRA demonstrate high performance and reliability in spite of the complexity of solved optimization problems. ANN-based classifiers developed in this way outperform many alternative methods on mentioned benchmark classification problems. The workability and usefulness of proposed meta-heuristic optimization algorithms are confirmed.**

Keywords—neural networks; classification; optimization; biology-inspired algorithms

I. INTRODUCTION

Classification is the problem of identifying to which of a set of categories a new instance belongs [1]. Classification problems have many applications, for example, computer vision, speech recognition, document classification, credit scoring, biological classification, etc.

Currently various algorithms for solving these problems are developed: linear classifiers such as Naïve Bayes classifier, decision trees, quadratic classifiers, etc. Researchers frequently use Artificial Neural Networks (ANN) for categorization [2]. There are various works in which this method was used and it was established that generally it is efficient and works successfully (for example, [3]).

The ANN models have three primary components: the input data layer, the hidden layer(s) and the output measure(s) layer. Each of these layers contains nodes, and these nodes are connected to nodes at adjacent layer(s). The hidden layer(s) contain(s) two processes: the weighted summation function and the transformation function. Both of these functions relate the values from the input data to the output measures. This is what we call "ANN's structure": the number of hidden layers, the number of nodes (neurons), and the type of the activation function on each node. Nodes in network are interconnected and each connection has a weight coefficient; the number of these coefficients depends on the solving problem (number of inputs) and the number of hidden layers and nodes. Thus, networks with a more or less complex structure usually have many weight coefficients which should be adjusted.

The weighted summation function is typically used in a feed forward/back propagation neural network model. But researchers proposed also other optimization methods for training neural networks which show good results as well (for example, [4]).

In this study the meta-heuristic called Co-Operation of Biology Related Algorithms (COBRA) [5] and its modification for solving optimization problems with binary variables were used for the neural networks design and the weight coefficients adjustment. In this research, ANN's structure wasn't fixed as in our previous work ([6]), i.e., both the networks' structure and weight coefficients were tuned. The COBRA meta-heuristic is based on the cooperation of five known nature-inspired algorithms (Particle Swarm Optimization (PSO) [7], Wolf Pack Search (WPS) [8], Firefly Algorithm (FFA) [9], Cuckoo Search Algorithm (CSA) [10] and Bat Algorithm [11]). The workability and reliability of COBRA for optimization problems with real-valued variables was shown in [5] on 28 benchmark functions with up to 50 variables and later confirmed in [6] on ANN weight coefficients adjustment with up to 110 real-valued variables.

The aim of this paper is to demonstrate the workability and usefulness of developed meta-heuristic on much harder optimization problems related to the ANN-based classifiers structure design and the weight coefficients adjustment.

The rest of the paper is organized as follows. In Section II the problem statement is presented. Then in Section III we describe proposed optimization techniques (COBRA and its binary modification). In Section IV the workability of the meta-heuristic is demonstrated with ANN-based classifiers design for four real world classification problems: two of

Research is fulfilled within governmental assignment of the Ministry of Education and Science of Russian Federation for Siberian State Aerospace University, project 140/14.

medical diagnostics and two of bank scoring problems. Conclusion section contains the results discussion and further research directions consideration.

II. PROBLEM STATEMENT

Tuning of neural networks' structure and weight coefficients is considered as the solving two unconstrained optimization problems: the first one with binary variables and the second one with real-valued variables. Type of variables depends on the representation of ANN's structure and coefficients.

First of all we set maximum number of hidden layers equals to 5 and maximum number of neurons on each hidden layer equals to 5, so maximum number of neurons is equal to 25. We could choose a larger number of layers and nodes, but our aim was to show that even network with a relatively small structure can show good results if it is tuned with effective optimization techniques. Each node is represented by a binary string of the length 4. If the string consists of zeros ("0000") then this node doesn't exist in ANN. So, whole structure of neural network is represented by binary string of the length 100 (25x4), each 20 variables represent one hidden layer. The number of input layers depends on problem in hand. ANN has one output layer.

We use following 15 activation functions for nodes:

1. $f(x) = 1/(1+exp(-x))$;
2. $f(x) = 1$;
3. $f(x) = th(x)$;
4. $f(x) = exp(-x^2/2)$;
5. $f(x) = 1 - exp(-x^2/2)$;
6. $f(x) = x^2$;
7. $f(x) = x^3$;
8. $f(x) = sin(x)$;
9. $f(x) = exp(x)$;
10. $f(x) = |x|$;
11. $f(x) = \{-1, x < -1; x, -1 \leqslant x \leqslant 1; 1, x > 1\}$;
12. $f(x) = \{0, x < -0.5; x+0.5, -0.5 \leqslant x \leqslant 0.5; 1, x > 0.5\}$;
13. $f(x) = 2/(1+exp(x)) - 1$;
14. $f(x) = 1/x$;
15. $f(x) = sign(x)$.

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 $0\times2^0 + 1\times2^1 + 1\times2^2 + 0\times2^3 = 6$ and for this neuron we use the sixth activation function from the list above.

Thus we use the optimization method for problems with binary variables (binary COBRA) for finding the best structure and the optimization method for problems with real-valued variables (original COBRA) for every structure weight coefficients adjustment.

III. OPTIMIZATION TECHNIQUES

A. Co-Operation of Biology Related Algorithms (COBRA)

The meta-heuristic of Co-Operation of Biology Related Algorithms (COBRA) was developed on the base of five well-known optimization methods such as Particle Swarm Optimization Algorithm (PSO) [7], Wolf Pack Search Algorithm (WPS) [8], Firefly Algorithm (FFA) [9], Cuckoo Search Algorithm (CSA) [10] and Bat Algorithm (BA) [11]. These algorithms are biology related optimization approaches originally developed for continuous variables space. They mimic collective behavior of corresponding animal groups that allows finding global optima of real-valued functions. Main reason for development of a new meta-heuristic was the inability to say which of above-listed algorithms is the best one or which algorithm should be used for solving any given optimization problem [5]. At the same time these algorithms are very like to each other. The idea is the use of the cooperation of these algorithms instead of any attempts to understand which one is the best for the current problem in hand.

The proposed approach consists in generating of five populations (one population for each algorithm) which are then executed in parallel cooperating with each other. Proposed algorithm is a self-tuning meta-heuristic. That's why we don't have to choose the population size for each algorithm. The number of individuals in each algorithm's population can increase or decrease depending on the fact of increasing or decreasing fitness value. If the fitness value wasn't improved during a given number of generations, then the size of all populations increases. And vice versa, if the fitness value was constantly improved, then the size of all populations decreases. Besides, each population can "grow" by accepting individuals removed from other population. Population "grows" only if its average fitness is better than the average fitness of all other populations. Thereby we can determine "winner algorithm" on each iteration/generation.

The result of this kind of competition allows presenting the biggest resource (population size) to the most appropriate (in the current generation) algorithm. This property can be very useful in case of hard optimization problem when, as it is known, there is no single best algorithm on all stages of the optimization process execution.

One of the most important driving forces of the suggested meta-heuristic is the migration operator that creates a cooperation environment for component algorithms. All populations communicate with each other: they exchange individuals in such a way that a part of the worst individuals of each population is replaced by the best individuals of other populations. It brings up to date information on the best achievements to all component algorithms and prevents their preliminary convergence to its own local optimum that improves the group performance of all algorithms.

The performance of proposed algorithm was evaluated on the set of benchmark problems from CEC'2013 competition

[5]. This set of benchmark function (namely there were 28 unconstrained real-parameter optimization problems) was given in [12]; there are also explanations about conducted experiments. Validation of COBRA was carried out for functions with 10, 30 and 50 variables. Exemplarily, results obtained by COBRA for functions with 50 variables are summarized in the Table I.

TABLE I. RESULTS OBTAINED BY COBRA FOR D=50

Func	Best	Worst	Mean	STD
1	1.25064e-010	2.03478	0.0398999	0.282119
2	2.31111e-008	0.000438373	2.88286e-005	6.94652e-005
3	0.0233248	0.116958	0.0255152	0.0129499
4	2.08957e-005	70.5335	1.96286	9.93911
5	1.65436e-007	0.000335373	8.67789e-005	8.85592e-005
6	0.00844481	0.27767	0.0594686	0.0669021
7	0.935338	16.5683	2.60003	2.82798
8	0.00594799	1.36323	0.0743616	0.222684
9	2.48243	2.85034	2.49154	0.0515319
10	0.857683	3.11696	1.33163	0.515766
11	1.75789e-006	8.4607	0.631801	1.67698
12	0.0810058	1.89318	0.21244	0.306151
13	0.0103392	1.83364	0.0912777	0.250786
14	0.683042	36.0939	7.47445	10.3556
15	0.181524	28.8378	1.52264	4.52404
16	0.817785	1.26899	0.886845	0.120543
17	0.370399	43.1483	3.83199	6.97121
18	9.35737	170.802	47.7118	40.6562
19	0.266479	10.0605	0.812531	1.4497
20	4.64541	6.36684	4.87292	0.330076
21	0.00198453	0.0276665	0.0102773	0.0072405
22	0.156301	10.2556	1.80134	2.40691
23	10.005	55.5646	28.5018	18.5944
24	0.00111885	57.877	3.66228	10.1655
25	221.006	222.839	221.466	0.466641
26	0.0109783	264.3	60.8755	106.646
27	200.158	295.23	226.768	36.5256
28	148.863	154.979	149.974	1.35671

* Best = best achieved result, Worst = worst achieved result, Mean = average result, STD = standard deviation

So, experiments showed that COBRA works successfully and is reliable on this benchmark. Results also showed that COBRA outperforms its component algorithms when dimension grows and more complicated problems are solved [5].

B. Binary modification of COBRA

As it was mentioned, all above listed algorithms (PSO, WPS, FFA, CSA and BA) were originally developed for continuous valued spaces. However many applied problems are defined in discrete valued spaces where the domain of the variables is finite. For this purpose the binary modification of COBRA was developed.

COBRA was adapted to search in binary spaces by applying a sigmoid transformation to the velocity component (PSO, BA) and coordinates (FFA, CSA, WPS) to squash them into a range [0, 1] and force the component values of the positions of the particles to be 0's or 1's.

The basic idea of this adaptation was taken from [13]; firstly it was used for PSO algorithm. It's known that in PSO each particle has a velocity [7], so binarization of individuals is conducted by the use of the calculation value of the sigmoid function which is also given in [13]:

$$s(v) = 1/(1+exp(-v)).$$

After that a random number from the range [0, 1] is generated and corresponding component value of particle's position is 1 if is smaller than $s(v)$ and 0 otherwise.

In BA each bat also has a velocity [11], that's why we can apply exactly the same procedure for the binarization of this algorithm. But in WPS, FFA and CSA [8, 9, 10] individuals have no velocities. For this reason, the sigmoid transformation is applied to position components of individuals and then a random number is compared with obtained value.

Just for the illustration of modification results we consider here six benchmark optimization problems from [14] (Rosenbrock's function, Sphere function, Ackley's function, Griewank's function, Hyper-Ellipsoidal function and Rastrigin's functions) that were used for testing new algorithm. Maximum number of function evaluations was equal to 100000 but calculations were stopped after reaching error value smaller than 0.001. Obtained results are presented in Table II.

TABLE II. RESULTS OBTAINED BY BINARY MODIFICATION OF COBRA

Func	D	APS	ANofFE	AFV	STD
1	2	31	740	0.000182069	0.282119
	3	68	3473	0.000188191	6.94652e-005
	4	80	6730	0.00579879	0.0129499
2	2	27	567	0.000236274	9.93911
	3	30	775	0.000150127	8.85592e-005
	4	32	916	0.000355086	0.0669021
3	2	32	1439	0.00019874	2.82798
	3	51	2046	0.00150713	0.222684
	4	62	3030	0.00126295	0.0515319
4	2	33	931	0.000209168	0.515766
	3	32	868	0.000191162	1.67698
	4	79	1710	0.000347666	0.306151

Func	D	APS	ANofFE	AFV	STD
5	2	30	899	0.00032841	0.00046868
	3	65	1332	0.000506847	0.00140048
	4	160	2258	0.00411721	0.158903
6	2	28	1734	180.0002	0.00018536
	3	36	3294	169.801	0.169149
	4	41	5462	159.2	0.279294

* D = dimension, APS = average population size, ANofFE = average number of function evaluations, AFV = average function value, STD = standard deviation

Experiments showed that the COBRA's binary modification works successfully and reliable but slower than original version of COBRA for the same problems with smaller success rate obtained [15] (results by original COBRA for the same test functions are given in the Table III).

TABLE III. RESULTS OBTAINED BY ORIGINAL VERSION OF COBRA

Func	D	APS	ANofFE	AFV	STD
1	2	20	263	0.000652238	0.00034669
	3	24	605	0.000750922	0.00029065
	4	27	757	0.000790054	0.00029793
2	2	20	284	0.000753919	0.00027206
	3	22	552	0.000783528	0.00029003
	4	27	932	0.000817905	0.00028088
3	2	29	867	0.000588745	0.00030714
	3	33	1470	0.000774339	0.00028261
	4	32	1604	0.000739637	0.00037221
4	2	20	202	0.000678884	0.00032022
	3	25	581	0.000749783	0.00028233
	4	28	1085	0.000756105	0.00028641
5	2	22	369	0.000806724	0.00014068
	3	22	574	0.000989866	5.426e-005
	4	20	263	0.000652238	0.00034669
6	2	28	885	0.000695163	0.00015934
	3	27	860	180.001	0.00027384
	4	40	2082	170.001	0.00024772

* D = dimension, APS = average population size, ANofFE = average number of function evaluations, AFV = average function value, STD = standard deviation

Such result was expected as the binary modification needs more computing efforts in continuous variables space and shouldn't be used instead of original COBRA. However, it can be recommend for solving optimization problems with the binary representation of solutions.

IV. EXPERIMANTAL RESULTS

In order to load developed optimization techniques with really hard task we chose four benchmark classification problems: bank scoring in Australia, bank scoring in Germany, Breast Cancer Wisconsin and Pima Indians Diabetes [16]. Our choice was conditioned by the circumstance that these problems were solved by other researchers many times with different methods. Thus there are many results obtained by alternative approaches that can be used for the comparison.

A. Bank scoring

Firstly two applied bank scoring problems were solved with ANN-based classifiers: bank scoring in Germany and in Australia. For Australian bank scoring problem, one has 14 attributes (6 numerical and 8 categorical), 2 classes, 307 examples of the creditworthy customers and 383 examples for the non-creditworthy customers. For German bank scoring problem one has 20 attributes (13 qualitative and 7 numerical), 2 classes, 700 records of the creditworthy customers and 300 records for the non-creditworthy customers. Both datasets were taken from [16].

From optimization view point, neural networks for these problems have from 175 till 225 real-valued variables for weight coefficients and 100 binary variables for structure. For the final weight coefficients adjustment (for the best obtained structure) we established maximum number of function evaluation equal to 10000.

Alternative algorithms for comparison as well as the way of the performance estimation are taken from [17]. Obtained results are demonstrated in Table IV where the portion of correctly classified instances from testing sets (%) is presented. So, for Australian bank scoring problem obtained results are better than for alternative classifiers from Table IV and for German bank scoring problem obtained results are the second best.

TABLE IV. CLASSIFIERS' PERFORMANCE COMPARISON FOR BANK SCORING PROBLEMS

Classifier	Scoring in Australia (%)	Scoring in Germany (%)
ANN+COBRA (this study)	90.75	79.74
ANN+COBRA ([16])	89.07	78.29
2SGP	90.27	80.15
C4.5	89.86	77.73
Fuzzy	89.10	79.40
GP	88.89	78.34
CART	87.44	75.65
LR	86.96	78.37
CCEL	86.60	74.60
RSM	85.20	67.70
Bagging	84.70	68.40
Bayesian	84.70	67.90
Boosting	76.00	70.00
k-NN	71.50	71.51

Authors' results in Table IV are averaged on 20 algorithms executions. Standard deviation for Australian bank scoring problem is equal to 1.166% and for German bank scoring problem it was equal to 2.267%. Here is an example of obtained structure for Australian bank scoring problem (5 hidden layers, five neurons on each layer):

1) First layer is (0011 0111 1101 1000 0011), i.e. neurons with the 3rd, 7th, 13th, 8th and 3rd activation functions.

2) Second layer is (1111 1110 1101 1011 0111), i.e., neurons with the 15th, 14th, 13th, 11th, and 7th activation functions.

3) Third layer is (0111 1110 1100 1101 1111), i.e., neurons with the 7th, 14th, 12th, 13th, and 15th activation functions.

4) Fourth layer is (0001 0011 0001 0111 1100), i.e., neurons with the 1st, 3rd, 1st, 7th, and 12th activation functions.

5) Fifth layer is (1011 0101 1101 0001 1111), i.e., neurons with the 11th, 5th, 13th, 1st and 15th activation functions.

As one can see, the classifier structure is rather heterogeneous.

In [6] the same problems were solved with one layer 5 neurons ANN based classifier, adjusted by original COBRA, with essentially worse results. Besides we solved these two problems with ANN-based classifiers with fixed structure (5 layers, 5 neurons, all with the sigmoidal activation function, on each level) adjusting weight coefficients with original COBRA. It could be considered as the rule of thumb for the choice of the ANN structure by a human end user. Obtained results (89.85 for Australian problem and 78.66 for German problem) are better than in [6] but worse than obtained with described approach and this difference is statistically significant that was proven with the Wilcoxon test. The approach suggested in this study requires 15-20% more computational efforts that authors consider as the acceptable payment for better results and getting rid of the permanent doubt of the correctness of the choice of the structure.

B. Medical Diagnostic

Next two medical diagnostic problems were solved as well in the same way for testing developed optimization algorithms: Breast Cancer Wisconsin Diagnostic and Pima Indians Diabetes [16]. For Breast Cancer Wisconsin Diagnostic one has 10 attributes (patient's ID that wasn't used for calculations and 9 categorical attributes which possess values from 1 to 10), 2 classes, 458 records of the patients with benign cancer and 241 records of the patients with malignant cancer. For Pima Indians Diabetes one has 8 attributes (all numeric-valued), 2 classes, 500 patients that were tested negative for diabetes and 268 patients that were tested positive for diabetes). Benchmark data for these problems also were taken from [16].

From optimization view point, for these problems one has from 145 till 150 real-valued variables for weight coefficients and 100 binary variables for the structure selection. For the final weight coefficients adjustment (for the best obtained structure) we again established the maximum number of function evaluations equals to 10000.

Obtained results are presented in Table V and Table VI where portion of correctly classified instances from testing sets is presented. There are in Table V and Table VI also results of other researchers used other approaches found in scientific literature [18, 19].

TABLE V. CLASSIFIERS' PERFORMANCE COMPARISON FOR BREAST CANCER PROBLEM

Author (year)	Method	Accuracy (%)
This study (2014)	ANN+COBRA	98.95
Authors results (2013) [15]	ANN+COBRA	98.16
Quinlan (1996)	C4.5	94.74
Hamiton et al. (1996)	RAIC	95.00
Ster, Dobnikar (1996)	LDA	96.80
Nauck and Kruse (1999)	NEFCLASS	95.06
Pena-Reyes, Sipper (1999)	Fuzzy-GA1	97.36
Setiono (2000)	Neuro-rule 2a	98.10
Albrecht et al. (2002)	LSA machine	98.80
Abonyi, Szeifert (2003)	SFC	95.57
Polat, Günes (2007)	LS-SVM	98.53
Guijarro-Berdias et al. (2007)	LLS	96.00
Karabatak, Cevdet-Ince (2009)	AR + NN	97.40
Peng et al. (2009)	CFW	99.50

TABLE VI. CLASSIFIERS' PERFORMANCE COMPARISON FOR PIMA INDIANS DIABETES PROBLEM

Author (year)	Method	Accuracy (%)
This study (2014)	ANN+COBRA	80.15
Authors results (2013) [15]	ANN+COBRA	79.83
H. Temurtas, N. Yumusak, F. Temurtas (2009)	MLNN with LM (10xFC)	79.62
	PNN (10xFC)	78.05
	MLNN with LM	82.37
	PNN	78.13
Mehmet Recep Bozkurt1, Nilüfer Yurtay, Ziynet Yılmaz1, Cengiz Sertkaya (2012)	PNN	72.00
	LVQ	73.60
	FFN	68.80
	CFN	68.00
	DTDN	76.00
	TDN	66.80
	Gini	65.97
	AIS	68.80
S. M. Kamruzzaman, Ahmed Ryadh Hasan (2005)	FCNN with PA	77.344
K. Kayaer., T. Yıldırım (2003)	GRNN	80.21

Author (year)	Method	Accuracy (%)
	MLNN with LM	77.08
L. Meng, P. Putten, H. Wang (2005)	AIRS	67.40

Authors' results are averaged on 20 algorithm executions. Standard deviation for Breast Cancer Wisconsin problem was equal to 0.3564% and for Pima Indians Diabetes problem it was equal to 1.2958%.

In [15] the same problems were solved with ANN-based classifiers that have fixed structure (3 layers, 3 neurons with the sigmoidal activation function on each layer) that gave worse results than obtained within current study. Moreover, we again used 5x5 ANN-based classifiers with fixed structure and sigmoidal activation functions to model the choice of the human non-expert user. Results are 98.19 for the cancer problem and 79.89 for the diabetes problem. Wilcoxon test shows no statistical difference between these results and results from [5], i.e. essential growth of ANN size did not produce the great positive effect on the classifier performance. However, the results obtained in this study are different with statistical significance that confirmed with Wilcoxon test.

V. CONCLUSION

In this paper we have described new meta-heuristic, called Co-Operation of Biology Related Algorithms, and firstly introduced its modification for solving optimization problems with binary variables. We illustrated the performance estimation of the proposed algorithms on the sets of test functions.

Then we used described optimization methods for automated design of ANN-based classifiers. Binary modification of COBRA was used for the classifier structure optimization and original COBRA was used for the weight coefficients adjustment both within structure selection process and for the final tuning of the best selected structure. This approach was applied to four real-world classification problems.

Solving these problems are equivalent to solving big and hard optimization problems where objective functions have many (up to 225) variables and are given in the form of computational program. Suggested algorithms successfully solved all problems designing classifiers with the competitive performance that allows us to consider the study results as the confirmation of the algorithms reliability, workability and usefulness in solving real world optimization problems.

Directions of the future research are heterogeneous: the development of modifications for constrained and multicriteria optimization, the improvement of the cooperation and competition scheme within the approach, the addition of other algorithms in the cooperation and the invention of the algorithms selection technique, the development of the modification for mixed optimization problems, etc.

REFERENCES

[1] T. G. Dietterich, Machine learning research: Four current directions. AI Mag. 18, 1997.

[2] C. M. Bishop, "Theoretical foundation of neural networks", Aston Univ., Neural computing research group, UK Tech. Rep. NCRG-96-024, 1996.

[3] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification", in Proc. of International Joint Conference on Artificial Intelligence, 2011, pp. 1237-1242.

[4] T. Sasaki, M. Tokoro, "Evolving Learnable Neural Networks under Changing Environments with Various Rates of Inheritance of Acquired Characters: Comparison between Darwinian and Lamarckian Evolution", Artificial Life, 5 (3), 1999, pp. 203-223.

[5] Sh. Akhmedova, E. Semenkin, "Co-Operation of Biology Related Algorithms", in Proc. of the IEEE Congress on Evolutionary Computation (CEC 2013), Cancún (México), June 20-23, 2013, pp. 2207-2214.

[6] Sh. Akhmedova, A. Shabalov, "Development and Investigation of Biologically Inspired Algorithms Cooperation Metaheuristic", in Proc. of the Genetic and Evolutionary Computation Conference Companion (GECCO'13), Amsterdam (The Netherlands), July 6–10, 2013, pp. 1417-1418

[7] J. Kennedy, R. Eberhart, "Particle Swarm Optimization", in Proc. of IEEE International Conference on Neural networks, IV, 1995, pp. 1942–1948.

[8] Chenguang Yang, Xuyan Tu and Jie Chen, "Algorithm of Marriage in Honey Bees Optimization Based on the Wolf Pack Search", in Proc. of the International Conference on Intelligent Pervasive Computing, 2007, pp. 462-467.

[9] X. S. Yang, "Firefly algorithms for multimodal optimization", in Proc. of the 5th Symposium on Stochastic Algorithms, Foundations and Applications, 2009, pp. 169–178.

[10] X. S. Yang, S. Deb, "Cuckoo Search via Levy flights", in Proc. of the World Congress on Nature & Biologically Inspired Computing, IEEE Publications, 2009, pp. 210-214.

[11] X. S. Yang, "A new metaheuristic bat-inspired algorithm", Nature Inspired Cooperative Strategies for Optimization, Studies in Computational Intelligence, Vol. 284, 2010, pp. 65-74.

[12] J. J. Liang, B. Y. Qu, P, N, Suganthan and A. G, Hernandez-Diaz, "Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization," Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China, and Technical Report, Nanyang Technological University, Singapore.

[13] J. Kennedy, R. Eberhart, "A discrete binary version of the particle swarm algorithm", in Proc. of the World Multiconference on Systemics, Cybernetics and Informatics, Piscataway, NJ, 1997, pp. 4104-4109.

[14] M. Molga, Cz Smutnicki, "Test functions for optimization need". 2005.

[15] Sh. Akhmedova, E. Semenkin, "New optimization metaheuristic based on co-operation of biology related algorithms", Vestnik. Bulletine of Siberian State Aerospace University, Vol. 4 (50), 2013, pp. 92-99.

[16] A. Frank, A. Asuncion. (2010). UCI Machine Learning Repository. Irvine, University of California, School of Information and Computer Science. Available: http://archive.ics.uci.edu/ml

[17] J.-J. Huang, G.-H. Tzeng, Ch.-Sh Ong, "Two-stage genetic programming (2SGP) for the credit scoring model", Applied Mathematics and Computation, 174, 2006, pp. 1039–1053.

[18] A. Marcano-Cedeño, J. Quintanilla-Domínguez, D. Andina, "WBCD breast cancer database classification applying artificial metaplasticity neural network", Expert Systems with Applications: An International Journal, vol. 38, issue 8, 2011, pp. 9573-9579.

[19] H. Temurtas, N. Yumusak, F. Temurtas, "A comparative study on diabetes disease diagnosis using neural networks," Expert Systems with Applications, vol. 36, no. 4, 2009, pp. 8610–8615.