ANN-based Classifiers Automatically Generated by New Multi-objective Bionic Algorithm

Shakhnaz Akhmedova and Eugene Semenkin
Siberian State Aerospace University, Krasnoyarsk, Russian Federation

Keywords: Neural Network Design, Multi-objective Optimization, Nature-inspired Algorithms, Medical Diagnostics.

Abstract: An artificial neural network (ANN) based classifier design using the modification of a meta-heuristic called Co-Operation of Biology Related Algorithms (COBRA) for solving multi-objective unconstrained problems with binary variables is presented. This modification is used for the ANN structure selection. The weight coefficients of the ANN are adjusted with the original version of COBRA. Two medical diagnostic problems, namely Breast Cancer Wisconsin and Pima Indian Diabetes, were solved with this technique. Experiments showed that both variants of COBRA demonstrate high performance and reliability in spite of the complexity of the optimization problems solved. ANN-based classifiers developed in this way outperform many alternative methods on the mentioned classification problems. The workability of the proposed meta-heuristic optimization algorithms was confirmed.

1 INTRODUCTION

A classification problem is a problem of identifying to which predefined group or class an object needs to be assigned based on a number of observed attributes related to that object. Many problems in business, science and industry can be treated as classification problems. Various intelligent information processing techniques exist for solving classification problems, with one of them being an artificial neural network (ANN).

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 which are connected to nodes at adjacent layer(s). Also there is an activation function on each node. Thus, the number of hidden layers, the number of nodes on each layer, and the type of activation function on each node determine the "ANN structure". Each connection between neurons has a weight coefficient; the number of these coefficients depends on the problem being solved and the number of hidden layers and neurons. Thus, the goal is to generate a neural network with a relatively simple structure which would effectively solve a given classification problem. Therefore, the modification of the collective bionic meta-heuristic called Co-operation of Biology Related Algorithms (COBRA) (Akhmedova and Semenkin, 2013(1)) for solving multi-objective optimization problems with binary variables (COBRA-bm) was used for selecting the ANN structure.

The weighted summation function for neurons is typically used in a feed-forward/back propagation network model. Yet it has been established that using other optimization methods for tuning the weight coefficients of a network can be more efficient (Sasaki and Tokoro, 1999). In this study the collective bionic meta-heuristic COBRA was used for the adjustment of the ANN weight coefficients.

Further, in Section 2 the problem statement is presented. Then in Section 3 the description of the proposed optimization techniques (COBRA and its modification for solving multi-objective problems with binary variables) is given. In Section 4 the workability of the meta-heuristics is demonstrated with ANN-based classifier design for two medical diagnostic classification problems: Breast Cancer Wisconsin and Pima Indians Diabetes. In the conclusion results are discussed and directions for further research are considered.

2 PROBLEM STATEMENT

The tuning of neural network structure and weight coefficients is considered as the solving of two unconstrained optimization problems: the first one is

Akhmedova S. and Semenkin E..
ANN-based Classifiers Automatically Generated by New Multi-objective Bionic Algorithm.
DOI: 10.5220/0005571603100317
In *Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2015), pages 310-317
ISBN: 978-989-758-122-9

a multi-objective problem with binary variables and the second one is a one-criterion problem with real-valued variables. The type of variables depends on the representation of the ANN structure and coefficients.

First of all let us assume that the maximum number of hidden layers is equal to m and that the maximum number of neurons on a hidden layer is equal to n. So, the maximum number of neurons in a network is equal to m×n. Frequently, a more complex neural network solves a given classification problem at least as well as a less complex network. However, the large number of weight coefficients, which depend on the number of the hidden layers and neurons, influence on the network adjustment process and later on the decision-making time. Therefore, the aim of this study was to develop an algorithm with an automated ANN-based classifier design with relatively simple structures which would effectively solve classification problems. Thus, the structure design of a network was considered as a multi-objective optimization problem: the first objective function was related to classification error and the second objective was related to the complexity of the structure, which was measured by the total number of neurons. Both objectives were minimized.

In this study m and n were equal to 5, so the maximum number of neurons was equal to 25. We could have chosen a larger number of layers and nodes, but our aim was to show that even a network with a relatively small structure can show good results if it is tuned with effective optimization techniques. Each node was represented by a binary string of length 4. If the string consisted of zeros ("0000") then this node did not exist in the ANN. So, the whole structure of the neural network was represented by a binary string of length 100 (25×4), and each 20 variables represented one hidden layer. The number of input layers depended on the problem in hand. The ANN has one output layer.

We used 15 different activation functions for nodes: sigmoidal, hyperbolic tangent, threshold function, linear function, etc. A list of the activation functions used are given below:

$$\begin{aligned} &f_1(x) = 1/(1+\exp(-x)); \\ &f_2(x) = 1; \\ &f_3(x) = th(x); \end{aligned} \tag{1}$$

$$\begin{aligned} &f_4(x) = \exp(-x^2/2); \\ &f_5(x) = 1-\exp(-x^2/2); \\ &f_6(x) = x^2; \\ &f_7(x) = x^3; \\ &f_8(x) = \sin(x); \\ &f_9(x) = \exp(x); \\ &f_{10}(x) = |x|; \\ &f_{11}(x) = \begin{cases} -1, x < -1 \\ x, -1 \le x \le 1; \\ 1, x > 1 \end{cases} \\ &f_{12}(x) = \begin{cases} 0, x < -0.5 \\ x+0.5, -0.5 \le x \le 0.5; \\ 1, x > 0.5 \end{cases} \\ &f_{13}(x) = 2/(1+\exp(x)) - 1; \\ &f_{14}(x) = 1/x; \\ &f_{15}(x) = sign(x). \end{aligned} \tag{2}$$

For determining which activation function will be used on a given node the integer that corresponds to its binary string was calculated; this integer was assigned as the number of the activation function.

Thus, we used the modification of the optimization method COBRA for multi-objective unconstrained problems with binary variables (COBRA-bm) for finding the best structure and the original version of COBRA for the adjustment of every structure weight coefficient. The approach COBRA-bm was developed with the use of Pareto optimality theory, so a set of different structures (non-dominated solutions) were obtained for every classification problem solved. The aforementioned set was considered as an ensemble of neural networks with a weighted averaging decision making scheme for inferring the ensemble decision (Jordan and Jacobs, 1994).

3 CO-OPERATION OF BIOLOGY RELATED ALGORITHMS

3.1 Original COBRA

The method for solving one-criterion unconstrained real-parameter optimization problems based on the cooperation of five nature-inspired algorithms such as Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995), the Wolf Pack Search (WPS) (Yang, Tu and Chen, 2007), the Firefly Algorithm (FFA) (Yang, 2009), the Cuckoo Search Algorithm (CSA) (Yang and Deb, 2009) and the Bat Algorithm (BA) (Yang, 2010) and called Co-Operation of

Biology Related Algorithms (COBRA) was introduced in (Akhmedova and Semenkin, 2013(1)). The basic idea of this approach consists in generating five populations (one population for each mentioned algorithm) which are then executed in parallel cooperating with each other.

The algorithm proposed in (Akhmedova and Semenkin, 2013 (1)) is a self-tuning meta-heuristic. Therefore there is no need to choose the population size for algorithms. The number of individuals in the population of each algorithm can increase or decrease depending on whether the fitness value is improving or not. If the fitness value was not 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. Additionally, each population can "grow" by accepting individuals removed from other populations. A population "grows" only if its average fitness is better than the average fitness of all other populations. Besides, 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.

The performance of the proposed algorithm was evaluated on the set of benchmark problems from the CEC'2013 competition in (Akhmedova and Semenkin, 2013 (1)). This set of 28 unconstrained real-parameter optimization problems was given in (Liang et al., 2012); there are also explanations about the conducted experiments. A validation of COBRA was carried out for functions with 2, 3, 5, 10, and 30 real variables. Experiments showed that COBRA works successfully and is reliable on this benchmark. Results also showed that COBRA outperforms its component algorithms when the dimension grows and more complicated problems are solved.

3.2 COBRA-bm

The binary modification of the algorithm COBRA, namely COBRA-b (Akhmedova and Semenkin, 2013 (2)), was modified for solving multi-objective optimization problems, so there was no necessity to modify component algorithms to solve optimization problems with binary variables. Development of the approach COBRA-bm for solving binary-parameter multi-objective optimization problems required the use of multi-objective versions of the above-listed component algorithms. So, all these techniques were extended to produce a Pareto optimal front directly: PSO and WPS by using the σ-method (Mostaghim and Teich, 2003) and the FFA (Yang, 2013), CSA (Yang and Deb, 2011) and BA (Yang, 2012) as suggested in corresponding papers.

Consequently, first of all a brief description of COBRA and its component-algorithms will be given and then COBRA-bm will be introduced.

3.2.1 Component Algorithms

Initially we will assume that the multiobjective optimization problem with K objective functions should be solved, namely all objectives should be minimized.

The original Particle Swarm Optimization algorithm (PSO) was discovered through simplified social model simulation (Kennedy and Eberhart, 1995). It is related to bird flocking, fish schooling and swarm theory. In the PSO algorithm, each individual uses information about the best position found by the whole swarm-population and the best position found by itself. However there is no single best position for multiobjective problems. That is why while solving multiobjective problems by PSO the procedure of choosing the best position for the particle has to be modified. For this purpose the σ-method (Mostaghim and Teich, 2003) was used. Firstly, the external archive S for nondominated solutions was generated. For each particle the σ-parameter was calculated. So for the i-th particle the current best position was found as follows: the particle in archive S whose σ-parameter is closest to the σ-parameter of the i-th particle was chosen as the best current position for the i-th particle, where closeness was measured by Euclidean distance.

The WPS algorithm was inspired by research on the social behaviour of a wolf pack; it simulates the hunting process of a pack of wolves (Yang, Tu and Chen, 2007). As in the PSO optimization tool, in the WPS approach each individual uses information about the best found position by the whole population for its movement in the search space. For modification of the Wolf Pack Search algorithm the same procedure was used as for the Particle Swarm Optimization method. Namely, an external archive of nondominated solutions was generated. And then the σ-method was applied to search for the current best wolf.

The Firefly algorithm (FFA) was inspired by the flashing behaviour of fireflies (Yang, 2009). In the FFA algorithm all fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex. For any two flashing fireflies, the less bright one will move towards the brighter one but if there is no brighter firefly than a particular one, then it will move randomly. The brightness of a firefly is affected or determined by the landscape of the objective function.

For multiobjective optimization the Firefly Algorithm was extended to produce a Pareto optimal front directly (Yang, 2013). After evaluation of the brightness or objective values of all the fireflies the comparison of each pair of fireflies was conducted. Then a random weight vector is generated (with the sum of components equal to 1), so that a combined best solution g^* can be obtained. This combined best solution g^* was used in order to fulfil random walks more efficiently. Also the i-th firefly was attracted by the j-th firefly (moved towards it) only if it was dominated by the j-th firefly. And finally the nondominated solutions are then passed onto the next iteration.

The Cuckoo Search Algorithm (CSA) is an optimization algorithm inspired by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds (of other species) (Yang and Deb, 2009). CSA uses three idealized rules. First of all, each cuckoo lays one egg at a time, and dumps its egg in a randomly chosen nest. Secondly, the best nests with a high quality of eggs will carry over to the next generations. And finally, the number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability p_a.

For simplicity, this last assumption can be approximated by the fraction p_a of the n nests being replaced by new nests (with new random solutions).

For multiobjective optimization problems with K different objectives the theory of Pareto optimality was used and the first and last rules were modified to incorporate the multiobjective idea (Yang and Deb, 2011). For multiobjective problems the first rule can be described as follows: each cuckoo lays K eggs at a time and dumps them in a randomly chosen nest, egg k corresponds to the solution to the k-th objective. And the last rule can be described in this way: each nest will be abandoned with a probability pa and a new nest with K eggs will be built according to the similarities or differences between the eggs; some random mixing can be used to generate diversity.

The Bat Algorithm (BA), which is the last component-method of COBRA, was inspired by research on the social behaviour of bats (Yang, 2010). The BA is based on the echolocation of bats that they use to detect prey, avoid obstacles, and locate their roosting crevices in the dark. For multiobjective optimization the Bat Algorithm was also extended to produce a Pareto optimal front (Yang, 2012). Firstly, an external archive of nondominated solutions is generated. Then, on each iteration, all objectives are combined into a single objective so that the Bat Algorithm is used for single objective optimization. After that the archive of non-dominated solutions is updated.

All mentioned algorithms PSO, WPS, FFA, CSA and BA were originally developed for continuous valued spaces. Binary modifications of these algorithms were employed by using the technique described in the study (Kennedy and Eberhart, 1997). Namely, they were adapted to search in binary spaces by applying a sigmoid transformation to the velocity component (PSO, BA) or coordinates (FFA, CSA, WPS) to squash them into a range [0, 1] and force the component values of the positions of the individuals to be 0's or 1's.

The basic idea of this adaptation was firstly used for the PSO algorithm (Kennedy and Eberhart, 1997). In PSO each particle has a velocity (Kennedy and Eberhart, 1995), so the binarization of individuals is conducted by the use of the calculation value of the sigmoid function which is also given in (Kennedy and Eberhart, 1997). After that a random number from the range [0, 1] is generated and the corresponding component value of the position of the particle is 1 if it is smaller than the sigmoid function value for that velocity and 0 otherwise.

In BA each bat also has a velocity (Yang, 2009), which is why exactly the same procedure for the binarization of this algorithm was applied. Yet in WPS, FFA and CSA individuals have no velocities. For this reason, the sigmoid transformation is applied to the position components of individuals and then a random number is compared with the obtained value.

Thus, at first all the mentioned bionic algorithms were adapted for solving unconstrained multi-objective real-parameter problems and then modified for solving optimization problems with binary variables.

3.2.2 Proposed Technique

The multiobjective modifications of the above-described bionic algorithms for solving unconstrained optimization problems with binary variables were used as component algorithms. For each component algorithm an external archive S_i (i = 1, …, 5) of non-dominated solutions was generated and a general external archive S was created. The solutions in all archives S_i were compared and solutions which were non-dominated among all of them were placed in the archive *S*.

The development of the multi-objective modification of optimization tool COBRA (COBRA-bm) required changes in the procedure of selecting the winning algorithm and in the migration operator. For the procedure of selecting the winning algorithm and migration operator on each stage of the COBRA-bm execution, K weight coefficients whose sum is equal to 1 were initialized randomly. Then all objectives were combined into a single objective (weighted sum of K objectives). This single objective was called "fitness" on the current stage. Therefore, the winning algorithm was determined by this fitness and for migration individuals were sorted according to the mentioned single objective.

To validate the proposed algorithm COBRA-bm, a subset of test multi-objective problems with convex, non-convex and discontinuous Pareto fronts was selected: Schaffer's Min–Min problem (SCH) (Schaffer, 1985), Kursawe problem (KUR) (Kursawe, 1990), Fonseca and Fleming problem (FAF) (Fonseca and Fleming, 1993), ZDT4 and ZDT6 problems (Zitzler, Deb and Thiele, 2000). The mentioned problems are defined as problems with real-parameter functions; the number of variables for them varied from 3 to 25. Each real-valued variable was represented by a binary string with a length of 10 bit. Thus, the number of binary variables varied from 30 to 250. For the component algorithms the number of individuals was equal to 50; and the number of iterations was equal to 1000. So in order to compare the performance of the proposed COBRA-bm with its components, the maximum number of function evaluations for COBRA-bm was established to be equal to 50000 (50×1000). The maximum number of Pareto optimal points in the external set was equal to 200. These settings are adopted from the papers (Yang, 2013), (Yang and Deb, 2011) and (Yang, 2012).

After generating Pareto points by COBRA-bm, the corresponding Pareto front was compared with the true front. We define the distance between the estimated Pareto front PF^e and its corresponding true front PF^t as follows:

$$E = \| PF^e - PF^t \|^2 \quad (3)$$

The results obtained by the components and COBRA-bm are summarized in Table 1 and Table 2.

Thus, the simulations for this subset of test functions show that the proposed approach COBRA-bm is an efficient algorithm for solving multi- objective binary optimization problems. It can deal with highly non-linear problems with diverse Pareto optimal sets and different problem dimensions. Also COBRA-bm outperforms its components; so it could be recommended for use instead of them.

Table 1: Summary of results for component algorithms.

Func	PSO	WPS	FFA	CSA	BA
SCH	4.812 e-007	5.083 e-007	4.626 e-007	4.270 e-007	6.099 e-007
KUR	3.017 e-004	3.968 e-004	2.559 e-004	3.229 e-004	4.052 e-004
FAF (3)	1.026 e-002	1.720 e-002	1.718 e-002	1.799 e-002	1.903 e-002
FAF (10)	1.089 e-002	7.808 e-003	1.144 e-002	1.686 e-002	2.657 e-002
FAF (20)	2.727 e-002	2.840 e-002	2.958 e-002	2.918 e-002	3.431 e-002
FAF (25)	3.681 e-002	3.404 e-002	3.298 e-002	3.032 e-002	5.238 e-002
ZDT4	3.550 e-003	4.009 e-003	4.049 e-003	4.254 e-003	2.572 e-003
ZDT6	1.014 e-003	1.508 e-003	8.204 e-004	7.112 e-004	6.776 e-004

Table 2: Summary of results for COBRA-bm.

Func	COBRA-bm
SCH	3.290e-007
KUR	2.071e-004
FAF (3)	2.042e-003
FAF (10)	6.869e-003
FAF (20)	1.193e-002
FAF (25)	2.370e-002
ZDT4	2.336e-003
ZDT6	6.614e-004

4 EXPERIMENTAL RESULTS

In order to load the developed optimization techniques with a really hard task we chose two benchmark classification problems: Breast Cancer Wisconsin and Pima Indians Diabetes. Our choice was conditioned by the circumstance 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.

For Breast Cancer Wisconsin Diagnostic there are 10 attributes (the patient's ID that was not used for calculations and 9 categorical attributes which possess values from 1 to 10), 2 classes, 458 records of patients with benign cancer and 241 records of patients with malignant cancer. For Pima Indians Diabetes there are 8 attributes (all numeric-valued), 2 classes, 500 patients that tested negative for diabetes and 268 patients that tested positive for diabetes). Benchmark data for these problems was taken from (Frank and Asuncion, 2010).

From the viewpoint of optimization, for these problems there are from 145 to 150 real-valued variables for weight coefficients and 100 binary variables for selecting the structure. For the structure selection of the neural network the maximum number of function evaluations was equal to 900. For the final weight coefficient adjustment (for the set of the best obtained structures) the maximum number of function evaluations was equal to 10000.

Table 3: Classifier performance comparison for Pima Indians Diabetes problem.

Author (year)	Method	Accuracy (%)
H. Temurtas et al. (2009)	MLNN with LM	82.37
K. Kayaer et al. (2003)	GRNN	80.21
This study (2015)	**ANN+COBRA-bm**	**80.17**
Akhmedova et al. (2014)	ANN+COBRA-b	80.15
Akhmedova et al. (2013)	ANN+COBRA	79.83
H. Temurtas et al. (2009)	MLNN with LM(10xFC)	79.62
H. Temurtas et al. (2009)	PNN	78.13
H. Temurtas et al. (2009)	PNN (10xFC)	78.05
S. M. Kamruzzaman et al. (2005)	FCNN with PA	77.34
M.R. Bozkurt et al. (2012)	DTDN	76.00
M.R. Bozkurt et al. (2012)	LVQ	73.60
M.R. Bozkurt et al. (2012)	PNN	72.00
L. Meng et al. (2005)	AIRS	67.40

Table 4: Classifier performance comparison for Breast Cancer Wisconsin problem.

Author (year)	Method	Accuracy (%)
Peng et al. (2009)	CFW	99.50
Akhmedova et al. (2014)	ANN+COBRA-b	98.95
This study (2015)	**ANN+COBRA-bm**	**98.80**
Albrecht et al. (2002)	LSA machine	98.80
Polat, Günes (2007)	LS-SVM	98.53
Akhmedova et al. (2014)	ANN+COBRA	98.16
Setiono (2000)	Neuro-rule 2a	98.10
Karabatak, Cevdet-Ince (2009)	AR + NN	97.40
Pena-Reyes, Sipper (1999)	Fuzzy-GA1	97.36
Ster, Dobnikar (1996)	LDA	96.80
Guijarro-Berdias et al. (2007)	LLS	96.00
Abonyi, Szeifert (2003)	SFC	95.57
Nauck and Kruse (1999)	NEFCLASS	95.06
Hamiton et al. (1996)	RAIC	95.00
Quinlan (1996)	C4.5	94.74

The obtained results are presented in Tables 3 and 4 where a portion of the correctly classified instances from testing sets is presented. There are in Tables 3 and 4 also results of other researchers and their approaches found in scientific literature (Marcano-Cedeno, Quintanilla-Domínguez and Andina, 2011) and (Temurtas, Yumusak and Temurtas, 2009).

The results of this study are averaged on 20 algorithm executions. Mostly only 2-4 networks were obtained as non-dominated solutions in one program run for each medical diagnostic problem. Here is an example of the obtained structure for the Breast Cancer Wisconsin problem (2 neural networks where the first has 5 hidden layers 10 neurons altogether and the second also has 5 hidden layers with the total number of neurons equal to 13).

- The first network structure: the first layer is (0100 0010), i.e. neurons with the 4th and 2nd activation functions; the second layer is (1100 0001), i.e. neurons with the 12th and 1st activation functions; the third layer is (0101 0100 0000), i.e. neurons with the 5th and 4th activation functions; the fourth layer is (0011 1100 0010), i.e. neurons with the 3rd, 12th and 2nd activation functions; the fifth layer is (0001), i.e. neuron with the 1st activation function;
- The second network structure: the first layer is (1000 0001), i.e. neurons with the 8th and 1st activation functions; the second layer is (0001 0100), i.e. neurons with the 1st and 4th activation functions; the third layer is (0100 0011 1101 0010), i.e. neurons with the 4th, 3rd, 13th and 2nd activation functions; the fourth layer is (0100 0010 0100), i.e. neurons with the 4th and 2nd activation functions; the fifth layer is (1000 0001), i.e. neurons with the 8th and 1st activation functions.

In (Akhmedova and Semenkin, 2014) the same problems were solved with ANN-based classifiers automatically generated by the one-criterion algorithms of COBRA and its binary modification COBRA-b which demonstrated good results but the networks designed were too complex. Experiments show non-significant statistical difference in the level of performance between the results obtained in this study and the results from (Akhmedova and Semenkin, 2014), i.e. essentially bigger ANNs did not produce a positive effect on the classifier performance.

5 CONCLUSIONS

In this paper a new meta-heuristic, called Co-Operation of Biology Related Algorithms, was described, and its modification COBRA-bm was introduced for solving multi-objective optimization problems with binary variables.

We illustrated the performance estimation of the proposed algorithms on subsets of test functions.

Then we used the described optimization methods for the automated design of ANN-based classifiers in two medicine diagnosis problems. The binary multi-objective modification of COBRA was used for the optimization of classifier structure and the original COBRA was used for the adjustment of weight coefficients both within the structure selection process and for the final tuning of the best selected structure.

This approach was applied to two real-world classification problems. Solving these problems are equivalent to solving big and hard optimization problems where objective functions have many (up to 150) variables and are given in the form of a computational program. The suggested algorithms successfully solved both problems with competitive performance that allows us to consider the study results as the confirmation of the reliability, workability and usefulness of the algorithms in solving real world optimization problems.

ACKNOWLEDGEMENTS

Research is performed with the financial support of the Ministry of Education and Science of the Russian Federation within the federal R&D programme (project RFMEFI57414X0126).

REFERENCES

Akhmedova, Sh., Semenkin, E., 2013(1). Co-Operation of Biology-Related Algorithms. In *IEEE Congress on Evolutionary Computations*. IEEE Publications.

Akhmedova, Sh., Semenkin, E., 2013(2). *New optimization metaheuristic based on co-operation of biology related algorithms*, Vestnik. Bulletine of Siberian State Aerospace University. Vol. 4 (50).

Akhmedova, Sh., Semenkin E., 2014. Co-Operation of Biology Related Algorithms Meta-Heuristic in ANN-Based Classifiers Design. In *IEEE World Congress on Computational Intelligence.* IEEE Publications.

Fonseca, C.M., Fleming, P.J., 1993. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In *The Fifth International Conference on Genetic Algorithms*.

Frank, A., Asuncion. A., 2010. *UCI Machine Learning Repository*. Irvine, University of California, School of Information and Computer Science. http://archive.ics.uci.edu/ml.

Jordan, M.I., Jacobs, R.A., 1994. Hierarchical Mixture of Experts and the EM Algorithm. *Neural Computation*, 6.

Kennedy, J., Eberhart, R., 1995. Particle Swarm Optimization. In *IEEE International Conference on Neural Networks.*

Kennedy, J., Eberhart, R., 1997. A discrete binary version of the particle swarm algorithm. In *World Multiconference on Systemics, Cybernetics and Informatics.*

Kursawe, F., 1990. *A variant of evolution strategies for vector optimization*. Parallel Problem Solving for Nature, H.P. Schwefel, R. Männer, Ed. Lecture Notes Computer Science, Springer-Verlag, Berlin, Vol. 496.

Liang, J.J., Qu, B.Y., Suganthan, P.N., Hernandez-Diaz, A.G., 2012. *Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization*. Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China, and Technical Report, Nanyang Technological University, Singapore.

Marcano-Cedeno, A., Quintanilla-Domínguez, J., Andina, D., 2011. *WBCD breast cancer database classification applying artificial metaplasticity neural network*. Expert Systems with Applications: An International Journal, vol. 38, issue 8.

Mostaghim, S., Teich, J., 2003. Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In *IEEE Swarm Intelligence Symposium*. IEEE Service Center.

Sasaki, T., Tokoro, M., 1999. *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).

Schaffer, J.D., 1985. Multiple objective optimization with vector evaluated genetic algorithms. In *The 1st International Conference on Genetic Algorithms*.

Temurtas, H., Yumusak, N., Temurtas, F., 2009. *A comparative study on diabetes disease diagnosis using neural networks*. Expert Systems with Applications, vol. 36, no. 4.

Yang, Ch., Tu, X., Chen, J., 2007. Algorithm of Marriage in Honey Bees Optimization Based on the Wolf Pack Search. In *International Conference on Intelligent Pervasive Computing*.

Yang, X.S., 2009 Firefly algorithms for multimodal optimization. In *The 5th Symposium on Stochastic Algorithms, Foundations and Applications*.

Yang, X.S., 2010. A new metaheuristic bat-inspired algorithm. *Nature Inspired Cooperative Strategies for Optimization*, Studies in Computational Intelligence. Vol. 284.

Yang, X.S., 2012. *Bat Algorithm for Multiobjective Optimization*. Bio-Inspired Computation, Vol. 3, no. 5.
Yang, X.S., 2013. *Multi-objective firefly algorithm for continuous optimization*. Engineering with Computers, Vol. 29, issue 2.
Yang, X.S., Deb, S., 2009. Cuckoo Search via Levy flights. In *World Congress on Nature & Biologically Inspired Computing*. IEEE Publications.
Yang, X.S., Deb, S., 2011. *Multi-objective cuckoo search for design optimization*. Computers & Operations Research, Vol. 40.
Zitzler, E., Deb, K., Thiele, L., 2000. *Comparison of multiobjective evolutionary algorithms: empirical results*. Evolutionary Computation, Vol. 8 (2).