UDC 004.032.26(045)
DOI:10.18372/1990-5548.64.14857

[1]V. M. Sineglazov,
[2]O. I. Chumachenko,
[3]D. M. Omelchenko

HYBRID NEURAL NETWORK OPTIMIZATION SYSTEM BASED ON ANT ALGORITHMS

[1,3]Aviation Computer-Integrated Complexes Department, Faculty of Air Navigation Electronics and Telecommunications, National Aviation University, Kyiv, Ukraine
[2]Technical Cybernetic Department, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute," Kyiv, Ukraine
E-mails: [1]svm@nau.edu.ua ORCID 0000-0002-3297-9060, [2]chumachecko@tk.kpi.ua ORCID 0000-0003-3006-7460, [3]maksymshevchenko01@gmail.com

Abstract—*The ant multi-criteria algorithm for feed forward neural networks training is proposed. It is used two criteria: the error of generalization and complexity. It is represented a review of neural network learning using swarm algorithms. As a result of training it is determined a structure of neural network (a number of layers and neurons in then) and the values of weight coefficients and biases. Modification of well-known algorithms consists in using the concept of Pareto optimality. It is done the research of proposed algorithm on the example of multilayer perceptron for the approximation problem solution.*

Index Terms—Multi-criteria optimization; ant algorithm; neural network; Pareto-optimality.

I. INTRODUCTION

Population algorithms involve the simultaneous processing of several options for solving the optimization problem and are an alternative to the classic "trajectory" search algorithms, in which in the field of search only one candidate for solving the problem is evolved.

The efficiency of such algorithms is equal to, and often higher than, the classical evolutionary algorithms, among which the most well-known is the genetic algorithm. With the help of population algorithms, complex optimization problems are successfully solved, for example, artificial neural networks (NNs) training.

The general idea of ant colony optimization (ACO) is the follow. Initially the ants wander randomly until food is found, and when an ant finds food; it returns to its nest depositing pheromones on its way back.

In nature, ants move on special paths due to the presence of a special substance on them – a pheromone, which is released by each ant during its movement. In ants, there are a large number of different pheromones for different signals, and now we are talking about the so-called trace pheromones. With their help, ants mark the path to the food source. If a wandering ant encounters such a source, it will emit a trace pheromone and go back to the anthill. Thus, other ants will follow this path. The more ants have come this way, the stronger the smell they leave and the more other ants it attracts. To enhance this effect, scientists have introduced a non-existent condition: pheromones evaporate the faster the longer the path traveled by the "ant". The result of this method is to find the shortest routes to all food sources, as well as the dynamic redistribution and optimization of these routes.

After investigating this behavior, with some adjustments, an ant algorithm was developed that is used to solve optimization problems, including learning about artificial NNs. This problem in the general formulation is called the problem of structural-parametric synthesis.

II. PROBLEM STATEMENT OF OPTIMAL ARTIFICIAL NEURAL NETWORK SYNTHESIS

Let us set the problem of NN synthesis as follows. First of all it's necessary to justify the criteria of the problem solution.

The complexity of the network mainly influences the generalization quality of the model. Too small network is not able to properly fit the true function described by the training data [14].

Another key point is computational efficiency. The network with a large number of hidden neurons which is clearly larger than necessary requires unneeded arithmetic calculations and in effect more computational resources. So it's necessary to use the network with less hidden neurons (less computational costs) because it is capable of reaching the same classification accuracy. A network with larger hidden layer has more weight connections that demands more dimension of weight-space, that creates better conditions for overcoming the local minima in the lower dimensional subspaces As a result, more paths are created around the barriers of poor local minima in the lower dimensional subspaces. Thus, local minima problem seemingly is intensified in case of too small networks. On the other hand depicts that many hidden neurons in the large network have very similar or identical hyperplanes. In fact, this similarity might result in redundancy of hidden neurons.

The complexity increases in an excessively large network, and even though it might be able to accurately approximate the desired function, nevertheless since a larger network learns quicker, it is more likely to have poor generalization due to over fitting. Consequently, learning process would be in further need of using various regularization techniques, which do not always lead to the best solutions.

So the problem statement can be determined as follow.

A finite set $J=\{(R_j, Y_j)\}, j=1,...,P$ pairs the "attribute value" type, where R_j, Y_j input and output vectors of NN, respectively.

It is necessary to synthesize such optimum NN based on trainning sample *J*, which would provided effective solution of the applied problem (classification, approximation, forecasting). Vector optimality criterion is defined as

$$I=\{I_1(x), I_2(x)\} \to \text{opt},$$

where $I_1(x)=E_{gen}(x)$ is the error of the generalization defining the solution error value of given task on the test sample; $I_2(x)=S(x)$ is the complexity of NN (number of inter neuronal connections); $X=(x_{11}, x_{12},...,x_{1x2}, x_2, x_3, x_4)$ is the vector that defines the topology, the structure and parameters of the network, where x_{1i} is the number of neurons in the *i*th hidden layer; $i=\overline{1, x_2}$, x_2 is the number of hidden layers; x_3 is the number of inter neuronal connections; x_4 is the set of w_{ij} weight coefficients values; *i*is the hidden layer number; *j* is the neuron number.

III. Review of Neural Network Learning using Swarm Algorithms for Multiobjective Optimization

In many real-world problems, candidate solutions are evaluated according to multiple, often conflicting objectives. Sometimes the importance of each objective can be exactly weighted, and hence objectives can be combined into a single scalar value by using, for example, a weighted sum. This is the approach used in [10] for a biobjective transportation problem. In other cases, objectives can be ordered by their relative importance in a lexicographical manner. In paper [1] it was proposed a two-colony ACS algorithm for the vehicle routing problem with time windows, where the first colony improves the primary objective and the second colony tries to improve the secondary objective while not worsening the primary one.

When there is no apriori knowledge about the relative importance of objectives, the goal usually becomes to approximate the set of Pareto-optimal solutions – a solution is Pareto optimal if no other solution is better or equal for all objectives and strictly better in at least one objective. In paper [2] it was considered various alternatives for extending ACO to multiobjective problems in terms of Pareto-optimality. They also tested a few of the proposed variants on a biobjective scheduling problem. In papers [3], [4] it was considered the application of ACO for multiobjective problems. Later studies have proposed and tested various combinations of alternative ACO algorithms for multiobjective variants of the QAP [5], [6], the knapsack problem [15], activity crashing [8], and the biobjective orienteering problem [9].

In paper [7] it was reviewed existing multiobjective ACO algorithms and carried out an experimental evaluation of several ACO variants using the bicriteria TSP as a case study. In paper [11] it was given another detailed overview of available multiobjective ACO algorithms.

IV. Problem Solution

A. General Approach

For combinatorial optimization problems ACO algorithms use the pheromone model to probabilistically construct solutions, the pheromone trail acts as the memory that stores the search experience of the algorithm. In ACO for continuous optimization the search is modeled with n variables, which is the number of NN weights in our case.

The candidate solutions are stored in an archive and are used to alter the probability distribution over the solution space. The probability distribution is analogous to the pheromone model used in combinatorial optimization problems. The algorithm maintains a solutions archive where are all the candidate solutions are stored; each candidate solution contains the values for all the '*n*' variables that define the search space. In this case '*n*' is equal to the number of weights in the NN that is trained. The archive contains '*k*' solutions. The fitness vector contains the result of the objective function $f(S_i)$ for all the solutions k in the archive. This is the function that we are trying to minimize. In this case we are trying to minimize the sum squared error for our training set. The archive is sorted against the values of this vector. The ACO algorithm performs the following tasks.

1) Initiates the archive with random values.

2) Creates and uses an instance of the NN object to determine the fitness of a solution.

3) Sorts the archive and generates biased candidate solutions by sampling a parameterized Gaussian random number generator.

4) Iterates the above process until the maximum number of iterations allowed is reached or if the training error criterion is satisfied.

B. Ant colony algorithm to optimize the feed forward network

For multioptimization task solution under adjustment of multilevel NN was used ACO [14], applied for single criterion task, as the base.

Firstly, ant colony algorithm is used to optimize the initial weights and network structure of feed forward NN. The algorithm is applied to train the network samples so that the network output error is minimized. The effective improvement of BP NN can easily fall into the local minimum value and the convergence speed is slow and all so defects. Suppose there are m parameters to be optimized of BP NN, the parameters are arranged in order: $p_1, p_2, \ldots, p_m$.

According to any parameter, initialization N any nonzero value, the set I_{pi} is constituted. Suppose the ant number is S, all ants search for food by randomly selected element from first set, then return to the nest after finding food. Repeat this step, until all ants collect the same route, the optimum solution is obtained of this network.

Modification of well-known algorithms consists in using the concept of Pareto optimality.

The specific calculation steps are as follows.

Step 1: The BP NN model is established, including the number of network layers, the number of nodes, the range and the sample size to be optimized.

Step 2: Initialize ant colony. The parameters are uniformly dispersed. Initialization the path that according to discretize data. Then the complete path is established. In initial time, the pheromone equals on all paths. The Pheromone trail intensity is $\tau_{ij}(0) = C_0$ (C_0 is constant) from the nest i to the food source j. The combination of discrete points of each parameter is defined as the path of ants that is a solution to the problem.

Step 3: Cyclic iteration of ant colony algorithm. In the course of the movement, the ant k ($k = 1, 2, \ldots, m$) decides the direction of the transfer according to the amount of information on each path. At the end of each iteration, generate random numbers q of the range from 0 to 1, make a comparison of explored relative importance threshold value $q_0 (0 \le q_0 \le 1)$ of the new path with the prior knowledge. If $q \le q_0$ the parameters of each weight are randomly variant according to the formula

$$C_{ij}^k = \begin{cases} \dfrac{\left(\tau_{ij}\eta_{ij}\right)^{\gamma}}{\sum\limits_{s \in allowed_k} \tau_{is}^{\alpha}(t)\eta_{is}^{\beta}(t)} & \tau_{ij} \in \max\{\tau_{is}\}, \quad s \in allowed_k, \\ \dfrac{\left(\tau_{ij}\right)^{\alpha}\eta_{ij}}{\sum\limits_{s \in allowed_k} \tau_{is}^{\alpha}(t)\eta_{is}^{\beta}(t)} & \tau_{ij} = \tau_{is} - \max\{\tau_{is}\}, \quad p \le p_m, \quad s \in allowed_k, \\ \dfrac{\tau_{ij}\left(\eta_{ij}\right)^{\beta}}{\sum\limits_{s \in allowed_k} \tau_{is}^{\alpha}(t)\eta_{is}^{\beta}(t)} & \tau_{ij} = \tau_{is} - \max\{\tau_{is}\}, \quad p > p_m, \quad s \in allowed_k, \\ 0 & otherwise. \end{cases}$$

where γ is a reverse exponential perturbation factor; $p_m \in (0,1)$ is random mutation rate; p abides to the uniform distribution of random variables (formula indicates that ants in an iterative process can choose a number of paths), random mutation is for new weight discrete points to get added to the collection S. If $q \ge q_0$, select the weight according to the formula

$$P_{ij}^k(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum\limits_{s \in allowed_k} \tau_{is}^{\alpha}(t)\eta_{is}^{\beta}(t)}, & j \in allowed_k \\ 0, & otherwise. \end{cases}$$

where $P_{ij}^k(t)$ is the transition probability of the ant k in the path i and the path j, $allowed_k = \{0, 1, \ldots, n-1\}$ is the next step to allow the ant k to choose the target; τ_{ij} is pheromone trail strength of edges (i, j); η_{ij} is heuristic factor for edge (i, j), P_{ij}^k is transition probability for the ant k, α, β are two parameters.

Step 4: All ants set out from set I_{pi}, follow the path rule to find element in order, finally the food source is found. The rule of route choice is: All ants k (1, 2, …, M) arbitrary choose jth with certain probability. The probability formula:

$$\Pr ab\left(t_j^k(I_{pi})\right) = \left(t_j(I_{pi})\right) / \sum_{u=1} t_u(I_{pi}).$$

Step 5: When all the ants are completed, we input training sample. According to the formula (amount of pheromone on each path)

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\rho\Delta\tau_{ij}(t,t+1),$$

$$\Delta\tau_{ij}(t,t+1)=\sum_{k=1}^{m}\Delta\tau_{ij}^{k}(t,t+1),$$

the weight parameters were updated. In the formula, $\Delta\tau_{ij}^{k}(t,t+1)$ is the pheromone of the ant k to stay on the path (i,j) at all times of $(t,t+1)$ The value depends on the merits and virtues of the ants. The shorter the path is, the more pheromone releases. $\Delta\tau_{ij}(t, t+1)$ is increment of the pheromone amount of the loop path (i, j), ρ is attenuation coefficient of pheromone. Usually, we set $\rho < 1$ to avoid an unlimited increase in the amount of path on the path. We will use the random perturbation strategy to prevent the stagnation of ant colony algorithm. The random selection probability needs to adjust dynamically [13], [14]. Approximate optimal solutions are obtained. The calculation time is shortened and the calculation efficiency is improved.

According to the Pareto optimization principles, $\Delta\tau_{ij}^{k}(t,t+1)$ is calculated depending on selection, based on multiple objectives (their fitness-functions). One of the possible approaches – tournament-based selection, which is described bellow.

Let $s_i, s_j, i, j \in [1:|S|], i \neq j$ be two comparable agents in the current positions X_i, X_j, where S is the set of agents. The more preferred one of these agents is determined by sequential comparison of pairs $(f_k(X_i), f_k(X_j))$ until it is established that $f_{k_i}(X_i) \le f_{k_j}(X_j); k, k_i \in [1:|F|]$. In this case, we assume that the agent s_i is fitted better, compared to the agent s_j.

The selection algorithm of preferred agent $s_{i_{best}} \in S$ can be described as follows:

1) We form a tournament – a Cartesian product of a set of agents with themselves: $T = S \times S$, where pairs of agents with themselves (s_i, s_i) are excluded.

2) For each pair $t_i \in T$ we compare the values of fitness functions $f_j(c_j)$ for each criterion c_j of agents of the pair $s_{i1}, s_{i2} \in t_i$; if one agent is better than the other one by the selected criterion – $f_j(c_{j_{s_{i1}}}) > f_j(c_{j_{s_{i2}}})$ – the first one is given +1 point to it's total score R_i; otherwise the score is assigned to the opposite agent.

The most adapted agent $s_{i_{best}} \in S, R_{i_{best}} \to \max$ is chosen.

Step 6: A set of the best weights that find by the ant colony algorithm are used as the initial weights of the adjusting algorithm. The error between the network output and the actual output is calculated. And the error is propagated from the output layer to the input layer, adjust weights, if the error reaches a predetermined precision or satisfy the maximum iterations number T, the algorithm is over. Otherwise, re-select ant colony to Step 2. The flowchart of an artificial NN based on ant colony optimization is described in Fig. 2.

C. *Number of ants* [12]

The selection of the ‘‘right’’ number of ants is a very critical issue affecting the performance of the algorithm. The number of ants must be sufficient to explore all potential states, while expending the least possible time. It should be noted that the ‘‘optimum’’ number of ants is specific to the data set(s) considered. Increasing the number of ants not only resulted in higher time requirements to reach a solution but also increased the testing error of ANN using the subset of features developed as solution.

D. *Number of generations*

Similarly, number of generations is an important parameter. Increasing the number of generations increases runtime of the algorithm tremendously while fewer generations make ants explore less possible states for each value of n, leading to poor/pre-mature convergence.

E. Simulation Results

Since our problem is an approximation, we will use the sigmoid as an activation function and the root mean square error (MSE) to calculate the error when constructing a NN:

$$f(s)=\frac{1}{1+e^{-x}},$$

$$L_{MSE}=\sqrt{\frac{\sum_{k=1}^{n}(x_k-x'_k)^2}{n}},$$

where x_k is the resulting value of the network; x'_k is the real value; n is the total number of sample set.

The stages of topology setup were presented in the corresponding Table I. The setup were performed for 100 generations of ants, the results are given with step of 10. The maximum number of possible layers is 5, the maximum number of neurons is 10. The results show the number of neurons in each layer. The last two columns contains the values of two criteria: accuracy and complexity, according to which the optimization was performed. The values of the criteria are calculated respectively, as

$$f_{loss} = \frac{1}{l}, \qquad f_{size} = 1 - \frac{n}{KN}.$$

The result of the adjustment was the following topology, represented as the coordinates of the ant: [2, 3, 2, 0, 0, 0], which can be deciphered as 2 hidden layers of 3 and 2 neurons, respectively.

TABLE I. STAGES OF TOPOLOGY SETUP

Generation	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Precision	Complexity
10	7	1	0	0	0	28.86	0.84
20	2	7	0	0	0	30.71	0.82
30	3	3	0	0	0	28.81	0.88
40	4	3	0	0	0	29.12	0.86
50	5	2	0	0	0	29.07	0.86
60	2	2	1	0	0	28.90	0.90
70	1	3	0	0	0	28.49	0.92
80	5	1	0	0	0	29.03	0.88
90	2	3	0	0	0	28,85	0.90
100	3	2	0	0	0	29.26	0.90

We compared the obtained topology with several examples of other topologies that could be used in this case: [5, 4, 2] – 3 layers of 5, 4 and 2 neurons, respectively; [4, 2]; [5, 3].

As we can see in the graph (Fig. 1) and from the Table II, in terms of accuracy, the topology obtained by a hybrid algorithm, in comparison with the examples, has a satisfactory accuracy.

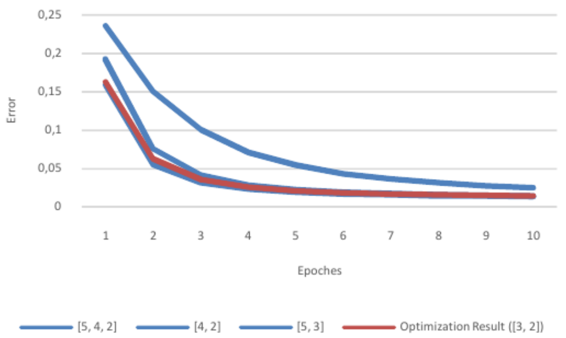

Fig. 1. Graphs of the average error of the first 10 epochs of study

TABLE II. AVERAGE ERRORS OF NEURAL NETWORKS IN THE 100TH EPIC

Topology	**Error**
[5, 4, 2]	0.012458
[4, 2]	0.011686
[5, 3]	0.012140

It should also be noted that the size of the obtained topology is the smallest among those presented, which means that the algorithm performs the optimization task as needed and that the described algorithm can be developed and used to solve practical problems.

V. CONCLUSIONS

It is proposed a multi-criteria algorithm for the feed forward NN structural-parametric synthesis It is shown the necessity of taking into account two criteria: accuracy and complexity. The use of proposed algorithm gave good results under approximation problem solution with help of multilayer perceptron.

REFERENCES

[1] L. M. Gambardella, E. D. Taillard, and G. Agazzi, MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows. In: Corne D, Dorigo M, Glover F, editors. New ideas in optimization. London: McGraw Hill; 1999, pp. 63–76.

[2] S. Iredi, D. Merkle, and M. Middendorf, “Bi-criterion optimization with multi colony ant algorithms,” in: Zitzler E, Deb K, Thiele L, et al., editors. Vol. 1993, *1st International Conference on Evolutionary Multi-Criterion Optimization, (EMO’01), Lecture Notes in Computer Science*. Heidelberg: Springer; 2001. pp. 359–372. https://doi.org/10.1007/3-540-44719-9_25

[3] K. F. Doerner, W. J. Gutjahr, R. F. Hartl, et al., “Ant colony optimization in multiobjective portfolio selection,” *in Proceedings of the Fourth Metaheuristics International Conference*; 2001. pp. 243–248.

[4] K. F. Doerner, W. J. Gutjahr, R. F. Hartl, et al., "Pareto ant colony optimization: a meta-heuristic approach to multiobjective portfolio selection," *Annals of Operations Research*, 2004;131:79–99. https://doi.org/10.1023/B:ANOR.0000039513.99038.c6

[5] Manuel López-Ibáñez, Luís Paquete and Thomas Stützle, “On the Design of ACO for the Biobjective Quadratic Assignment Problem,” in Dorigo M., et al., editors, vol. 3172, *Ant Colony Optimization and Swarm Intelligence: 4th International Workshop, ANTS 2004, Lecture Notes in Computer Science. Heidelberg*: Springer; 2004. pp. 214–225. https://doi.org/10.1007/978-3-540-28646-2_19

[6] Manuel López-Ibáñez, Luís Paquete, and Thomas Stützle, “Hybrid population-based algorithms for the bi-objective quadratic assignment problem,” *J Math Model Algorithms*, 2006, 5(1): 111–137. https://doi.org/10.1007/s10852-005-9034-x

[7] C. Garcrıa-Martırnez, O. Cordorn, and F. Herrera, “A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP,” *Eur J Oper Res*, 2007; 180(1):116–148. https://doi.org/10.1016/j.ejor.2006.03.041

[8] K. F. Doerner, W. J. Gutjahr, R. F. Hartl, et al., "Nature-inspired metaheuristics in multiobjective activity crashing," *Omega*, 2008, 36(6):1019–1037. https://doi.org/10.1016/j.omega.2006.05.001

[9] M. Schilde, K. F. Doerner, R. F. Hartl, et al., "Metaheuristics for the bi-objective orienteering problem," *Swarm Intell*, 2009, 3(3):179–201. https://doi.org/10.1007/s11721-009-0029-5

[10] K. F. Doerner, R. F. Hartl, and M. Reimann, "Are CompetANTS more competent for problem solving? The case of a multiple objective transportation problem," *Cent Eur J Oper Res Econ*, 2003, 11(2):115–141.

[11] D. Angus, and C. Woodward, "Multiple objective ant colony optimization," *Swarm Intell*, 2009, 3(1):69–85. https://doi.org/10.1007/s11721-008-0022-4

[12] Rahul Karthik Sivagaminathan and Sreeram Ramakrishnan, "A hybrid approach for feature subset selection using neural networks and ant colony optimization," *Expert Systems with Applications: An International Journal*, 33 (2007) 49–60. https://doi.org/10.1016/j.eswa.2006.04.010

[13] F. Wan, F. Q. Wang, and W. L. Yuan, "The reservoir runoff forecast with artificial neural network based on ant colony optimization," *Applied ecology and environmental research* 15(4): 497–510. https://doi.org/10.15666/aeer/1504_497510

[14] Saman Sadeghyan and Shahrokh Asadi, *MS-BACO: A new model selection algorithm using binary ant colony optimization for neural complexity and error reduction*. Published 2018, Computer Science.

[15] I. Alaya, C. Solnon, and K. Gherdira, "Ant colony optimization for multi-objective optimization problems," vol. 1, *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007). Los Alamitos (CA)*: IEEE Computer Society Press; 2007, pp. 450–457. https://doi.org/10.1109/ICTAI.2007.108

Received April 24, 2020.

Sineglazov Victor. orcid.org/0000-0002-3297-9060
Doctor of Engineering Science. Professor. Head of the Department.
Aviation Computer-Integrated Complexes Department, Faculty of Air Navigation Electronics and Telecommunications, National Aviation University, Kyiv, Ukraine.
Education: Kyiv Polytechnic Institute, Kyiv, Ukraine, (1973).
Research area: Air Navigation, Air Traffic Control, Identification of Complex Systems, Wind/Solar power plant.
Publications: more than 600 papers.
E-mail: svm@nau.edu.ua

Chumachenko Olena. orcid.org/0000-0003-3006-7460. Doctor of Engineering Science. Associate Professor.
Technical Cybernetic Department, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute," Kyiv, Ukraine.
Education: Georgian Polytechnic Institute, Tbilisi, Georgia, (1980).
Research area: system analysis, artificial neuron networks.
Publications: more than 80 papers.
E-mail: chumachecko@tk.kpi.ua

Omelchenko Dmytro. Bachelor.
Aviation Computer-Integrated Complexes Department, Faculty of Air Navigation Electronics and Telecommunications, National Aviation University, Kyiv, Ukraine.
Education: National Aviation University, Kyiv, Ukraine, (2020).
Research area: Identification of Complex Systems.
E-mail: maksymshevchenko01@gmail.com

В. М. Синєглазов, О. І. Чумаченко, Д. М. Омельченко. Система оптимізації гібридної нейронної мережі на основі мурашиного алгоритму

Запропоновано багатокритеріальний мурашиний алгоритм для навчання нейронних мереж прямого поширення. Використовується два критерії: помилка узагальнення і складність. Представлено огляд методів навчання нейронної мережі з використанням ройових алгоритмів. В результаті навчання визначається структура нейронної мережі (кількість шарів і нейронів у ній) та значення вагових коефіцієнтів і зсувів. Модифікація відомих алгоритмів полягає у використанні концепції оптимальності за Парето. Проведено дослідження запропонованого алгоритму на прикладі багатошарового персептрона для розв'язання задачі апроксимації.

Ключові слова: багатокритеріальна оптимізація; мурашиний алгоритм; нейронна мережа; Парето-оптимальність.

Синєглазов Віктор Михайлович. orcid.org/0000-0002-3297-9060
Доктор технічних наук. Професор. Завідувач кафедрою.
Кафедра авіаційних комп'ютерно-інтегрованих комплексів, Факультет аеронавігації електроніки і телекомунікацій, Національний авіаційний університет, Київ, Україна.
Освіта: Київський політехнічний інститут, Київ, Україна, (1973).

Напрям наукової діяльності: аеронавігація, управління повітряним рухом, ідентифікація складних систем, вітроенергетичні установки.
Кількість публікацій: більше 600 наукових робіт.
E-mail: svm@nau.edu.ua

Чумаченко Олена Іллівна. orcid.org/0000-0003-3006-7460. Доктор технічних наук. Доцент.
Кафедра технічної кібернетики, Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського», Київ, Україна.
Освіта: Грузинський політехнічний інститут, Тбілісі, Грузія, (1980).
Напрямок наукової діяльності: системний аналіз, штучні нейронні мережі.
Кількість публікацій: понад 80 наукових робіт.
E-mail: chumachecko@tk.kpi.ua

Омельченко Дмитро Валерійович. Бакалавр.
Кафедра авіаційних комп'ютерно-інтегрованих комплексів, Факультет аеронавігації електроніки і телекомунікацій, Національний авіаційний університет, Київ, Україна.
Освіта: Національний авіаційний університет, Київ, Україна, (2020).
Напрям наукової діяльності: ідентифікація складних систем.
E-mail: maksymshevchenko01@gmail.com

В. М. Синеглазов, Е. И. Чумаченко, Д. М. Омельченко. Система оптимизации гибридной нейронной сети на основе муравьиного алгоритма

Предложен многокритериальный муравьиный алгоритм для обучения нейронных сетей прямого распространения. Используется два критерия: ошибка обобщения и сложность. Представлен обзор методов обучения нейронной сети с использованием роевых алгоритмов. В результате обучения определяется структура нейронной сети (количество слоев и нейронов в ней) и значения весовых коэффициентов и смещений. Модификация известных алгоритмов заключается в использовании концепции оптимальности по Парето. Проведено исследование предложенного алгоритма на примере многослойного персептрона для решения задачи аппроксимации.

Ключевые слова: многокритериальная оптимизация; муравьиный алгоритм; нейронная сеть; Парето-оптимальность.

Синеглазов Виктор Михайлович. orcid.org/0000-0002-3297-9060
Доктор технических наук. Профессор. Заведующий кафедрой.
Кафедра авиационных компьютерно-интегрированных комплексов, Факультет аэронавигации электроники и телекоммуникаций, Национальный авиационный университет, Киев, Украина.
Образование: Киевский политехнический институт, Киев, Украина, (1973).
Направление научной деятельности: аэронавигация, управление воздушным движением, идентификация сложных систем, ветроэнергетические установки.
Количество публикаций: более 600 научных работ.
E-mail: svm@nau.edu.ua

Чумаченко Елена Ильинична. orcid.org/0000-0003-3006-7460. Доктор технических наук. Доцент.
Кафедра технической кибернетики, Национальный технический университет Украины «Киевский политехнический институт им. Игоря Сикорского», Киев, Украина.
Образование: Грузинский политехнический институт, Тбилиси, Грузия, (1980).
Направление научной деятельности: системный анализ, искусственные нейронные сети.
Количество публикаций: более 80 научных работ.
E-mail: chumachecko@tk.kpi.ua

Омельченко Дмитрий Валерьевич. Бакалавр.
Кафедра авиационных компьютерно-интегрированных комплексов, Факультет аэронавигации электроники и телекоммуникаций, Национальный авиационный университет, Киев, Украина.
Образование: Национальный авиационный университет, Киев, Украина, (2020).
Направление научной деятельности: идентификация сложных систем.
E-mail: maksymshevchenko01@gmail.com