

2-28-2019

Neuroevolutional Methods for Decision Support Under Uncertainty

Nina Komleva

Plekhanov Russian University of Economics - Moscow

Olga Khlopkova

Plekhanov Russian University of Economics - Moscow

Matthew He

Nova Southeastern University, hem@nova.edu

Follow this and additional works at: https://nsuworks.nova.edu/math_facarticles

 Part of the [Computer Sciences Commons](#), and the [Mathematics Commons](#)

NSUWorks Citation

Komleva, Nina; Khlopkova, Olga; and He, Matthew, "Neuroevolutional Methods for Decision Support Under Uncertainty" (2019).
Mathematics Faculty Articles. 260.
https://nsuworks.nova.edu/math_facarticles/260

This Article is brought to you for free and open access by the Department of Mathematics at NSUWorks. It has been accepted for inclusion in Mathematics Faculty Articles by an authorized administrator of NSUWorks. For more information, please contact nsuworks@nova.edu.



NEUROEVOLUTIONAL METHODS FOR DECISION SUPPORT UNDER UNCERTAINTY

Komleva N., Ph.D.

Plekhanov Russian University of Economics, Moscow, Russia

Khlopkova O., Ph.D.

Plekhanov Russian University of Economics, Moscow, Russia

Matthew He, Ph.D.

Halmos College of Natural Sciences and Oceanography
Nova Southeastern University, Florida, USA

ABSTRACT

The article presents a comparative analysis of the fundamental neuroevolutional methods, which are widely applied for the intellectualization of the decision making support systems under uncertainty. Based on this analysis the new neuroevolutionary method is introduced. It is intended to modify both the topology and the parameters of the neural network, and not to impose additional constraints on the individual. The results of the experimental evaluation of the performance of the methods based on the series of benchmark tasks of adaptive control, classification and restoration of damaged data are carried out. As criteria of the methods evaluation the number of failures and the total number of evolution epochs are used.

Keywords: neuroevolution, evolutionary modeling, neural networks, genetic algorithms, computational intelligence.

Cite this Article: Komleva N., Ph.D., Khlopkova O., Ph.D. and Matthew He, Ph.D., Neuroevolutional Methods for Decision Support under Uncertainty, *International Journal of Civil Engineering and Technology*, 10(2), 2019, pp. 1263-1276.

<http://www.iaeme.com/IJCIET/issues.asp?JType=IJCIET&VType=10&IType=02>

1. INTRODUCTION

With the development of the information society, the trend towards an increase in information flows in the overwhelming majority of organizations is becoming more apparent [1,2]. The number of virtual libraries, storage of multimedia content, as well as educational institutions with access to the Internet increases annually. There is a need for automated processing of content in information environments of any scale, ranging from information

and educational systems in universities and ending with corporate environments for project management. Automating the processing of huge amounts of data allows you to improve workflow, streamline and systematize information flows, and as a result - reduce labor and cost costs [3,4]. Moreover, the effectiveness and efficiency of the automated process depends entirely on the chosen method of content processing.

Research in the field of computational intelligence as a tool for the intellectualization of data processing is becoming increasingly practical each year. Computational intelligence (CI) is a branch of artificial intelligence and an alternative to its classical approaches based on rigorous inference. In the modern understanding, artificial intelligence (AI) is a set of methods and tools for solving complex applied problems that use the principles embodied in the processes occurring in living or inanimate nature (for example, in thinking). CI relies on heuristic, inaccurate, and approximate algorithms for solving problems that often have no solution in polynomial time. It combines the methods of training, adaptation, evolution and fuzzy logic to create a certain degree of intellectual programs.

Thus, computational intelligence works with a class of problems in the field of artificial intelligence, but classical AI solves them using traditional hard computing methods, and CI solves them using soft computing.

Soft computing is a symbiosis of computational methodologies that collectively provide the basis for understanding, designing and developing intelligent systems. Compared to hard computing, soft computing is more suited to processing inaccurate, uncertain, and incomplete data to achieve ease of manipulation, robustness, lower cost solutions, and better agreement with reality. CI tools are based on the following most actively developing approaches and soft computing methods: fuzzy systems (fuzzy sets, fuzzy logic, fuzzy regulators), neural networks, evolutionary modeling (genetic programming, evolutionary programming, genetic algorithms), chaos theory, swarm intelligence, artificial immune systems, hybrid systems (fuzzy neural networks, combinations of genetic algorithms and neural networks).

Computational intelligence provides a new perspective on statistical methods, as exemplified by fuzzy systems. Fuzzy logic underlies the methods of working with inaccuracy, approximate reasoning, and computing with words. Artificial neural networks, in turn, are closely related to machine learning. Neurocomputing reflects the ability to learn, adapt, and identify. In the case of genetic computations and swarm intelligence, we are talking about the ability to systematize a random search and achieve the optimum value of the characteristics of the object. Probabilistic calculations provide the basis for managing uncertainty and conducting reasoning based on evidence.

The basis for the development of computational intelligence is the development of algorithmic models for solving complex problems of high dimension by simulating natural systems. Each of the CI paradigms is based on the simulation of real-world biological phenomena and properties. Neural networks originated as a model of fragments of the nervous system; evolutionary calculations mimic natural evolution and use the laws of genetics; swarm intelligence simulates the social behavior of colonies of organisms; artificial immune systems would not be built without biological prototypes; fuzzy systems have arisen as a result of studying the interaction of organisms with the environment.

So, computational intelligence includes the following basic paradigms:

- Artificial neural networks
- Fuzzy systems
- Evolutionary modeling
- Swarm intelligence.

In addition, as part of computational intelligence, studies are conducted on artificial immune systems, chaos theory, fractal transformations, and wavelets.

A neural network is a distributed parallel processor consisting of elementary units of information processing — neurons that accumulate experimental knowledge.

A fuzzy system is a system with fuzzy: specification of parameters, descriptions of input and output variables, and a conclusion based on a knowledge base consisting of fuzzy production rules of the form IF (antecedent) THEN (consequent). A fuzzy system consists of a knowledge base, fuzzifier, term functions, fuzzy inference kernel, and in most cases defuzzifier.

As practice shows, fuzzy control gives better results in comparison with those obtained using classical algorithms [5]. The obvious area of implementation of fuzzy logic is all sorts of expert systems, including: nonlinear control and management in production, self-learning systems, systems recognizing texts in natural language, planning and forecasting systems based on incomplete information, financial analysis in conditions of uncertainty, database management data, improving management strategies and coordinating actions [6,7].

Computational intelligence methods are extremely diverse and have borrowed a great deal from biology, neurophysiology, genetics, sociology and psychology.

The specificity of CI is that they use probabilistic, rather than deterministic, rules for generating solutions, they allow you to accumulate and use knowledge about the search space and, therefore, show the ability to learn.

Throughout the history of computational intelligence, the consolidation of its methods is regularly observed. Thus, at present, the terms “evolutionary programming”, “evolutionary strategies”, “genetic algorithms” and “genetic programming” are considered as particular branches of the general approach of “evolutionary modeling”, sometimes called “evolutionary computations”. Neural network systems are increasingly integrated with fuzzy inference algorithms, genetic algorithms or methods of swarm intelligence. This allows you to increase the speed and quality of training neural networks, make them more dynamic and reliable.

Hybridization of intellectual processing of information is the motto of recent years in the field of artificial intelligence technologies. In particular, the methods of fuzzy logic in conjunction with the concept of neural networks have proven to be one of the most effective and promising tools for processing content.

This is due to the fact that the application of methods from the classical branches of mathematics, such as linear modeling, automatic control, decision support, in some problems is not optimal. Uncertainty of the mathematical description, the absence of a formal model and non-triviality are the properties of problems for which classical methods give inadequate results.

In principle, such methods are not intended to solve problems of intellectual processing and analysis of data. Neuro-fuzzy models are successfully used as their replacement. There is an extensive experience of their implementation and use in various subject areas, including the formulation of medical diagnoses, autopilot training, radar signal recognition, information filtering, monitoring and prevention of emergency situations, analysis of seismic activity, etc.

Neuroevolution is a dynamically developing computational intelligence discipline dealing with the study and development of hybrid methods for the design of neural networks by using evolutionary algorithms [8]. The primary goal of integration of neuroevolutionary modules into the decision making support systems is that it assists a decision maker to plan activities according to varying environment conditions.

The need for the intelligent support of decision making acutely arises especially when solving hard-to-formalize problems, as well as problems of any informal, creative and intellectual kind [9, 10]. First of all, such problems are characteristics of application domains not containing any functionally complete mathematical theories that describe the decision making objects and models [11]. As techniques of automatizing the decision making processes in such application domains, various neuroevolutionary methods are used with ever increasing frequency. The most common of them are the CE, NEAT, EANT, ENS³ and DXNN methods (Fig. 1) as illustrated in Section 2.

This article presents a comparative analysis of the fundamental neuroevolutionary methods in Section 2 and, based on this analysis, suggests a novel method KHO that allows modifying the topology and the parameters of a neural network, not imposing any additional constraint on that neural network in Section 3. The results of solving the series of traditional benchmark tasks by using the analyzed neuroevolutionary methods and the proposed method are also presented in Section 4. The aforesaid tasks are used in the neuroevolution field to make an indirect analysis of efficiency, performance, reliability and other parameters of the methods. The benchmark tasks serve as a criterion for selection of one or another method to be used in different practical application fields, and the successful passing of the series of benchmark tests is evidence of stable performance of the method.

2. COMPARATIVE ANALYSIS OF NEUROEVOLUTIONARY METHODS

At present, in spite of rapid development of the neuroevolution theory, the number of effective methods, which are expedient to be used in practice, is small. Let us consider the main of them.

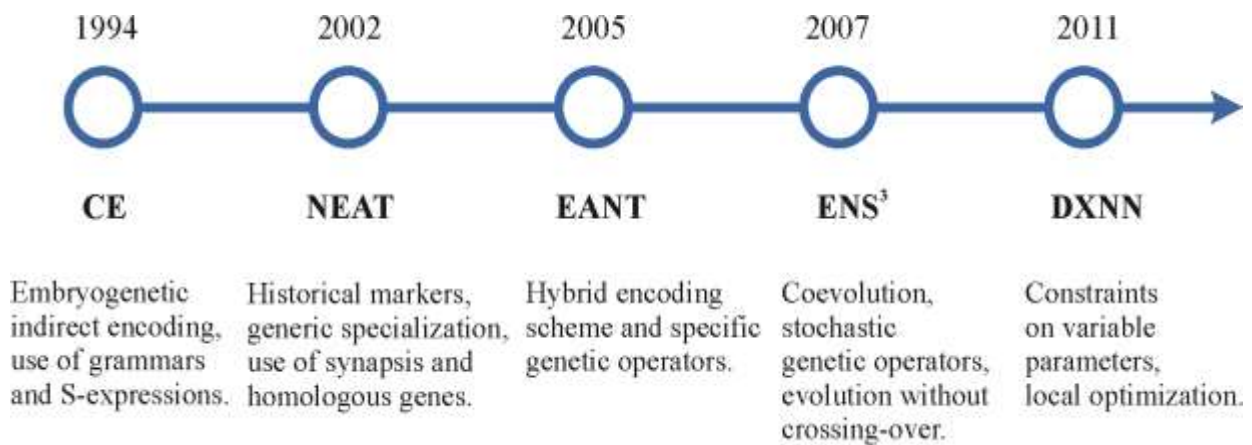


Figure 1 Timeline of Development of Main Neuroevolutionary Methods

CE (Cellular Encoding) – the indirect encoding method aimed at evolution of the sequence of using the rules that control the division of cells from which a neural network is produced [12]. The CE method is intended for parallel modification of the topology by way of sequentially complexifying and setting weight of the neural network.

The CE method is primarily oriented to the construction of modular neural networks consisting of hierarchically linked subnetworks. It is also useful for formation of patterns and recursive structures. An advantage of this method is the possibility of making changes in the neuron activation function. Also, the method makes it possible to produce a neural network of any configuration, without constraints on the number of neurons and the topology. In this

case, the individuals formed as a result of the use of genetic operators are guaranteed to be viable. A disadvantage of the method is high resource intensity because each cell stores a copy of the grammar tree, as well as the markers and internal registers. Since the method implements the indirect encoding, it is characterized by low efficiency resulting from the necessity to carry out the grammar tree encoding-decoding operations.

DXNN (Deus Ex Neural Network) – the memetic algorithm based method for separate modification of the topology and weights of a neural network [13]. Depending on its implementation, the DXNN method supports the direct and indirect chromosome encoding techniques. For evolution of the neural network topology, the method provides for a global search stage, while at a local search stage it optimizes the synaptic weights only.

The memetic approach implemented in the DXNN method has a number of advantages. The sequential modification of the topology and weights allows determining whether the given individual demonstrates low fitness due to an unsuccessfully formed topology or due to incorrectly selected weights. Also, in most neuroevolutionary methods, the operators involved in changing the weight values are applied indiscriminately to all neurons of the neural network, thus making the probability of optimizing the new and the right neuron, very low. In fact, the memetic methods, especially DXNN, optimize the weights of recently modified neurons, not affecting the architecture already optimized during previous iterations.

At the same time, the DXNN method is characteristic of some disadvantages: the evolution follows the path of complexifying the topology and increasing the number of neurons because the method does not realize such mutation operators as removal of the link or removal of the neuron; and of all the neuron parameters, the method optimizes the weights only. In this connection, the optimum solution time increases exponentially at a linear increase of task complexity.

EANT (Evolutionary Acquisition of Neural Topologies) – the method using a hybrid encoding scheme such as CGE (common genetic encoding) [14]. The CGE scheme is intended for separate evolution of both the structure and parameters of neural networks, and is characterized by possessing two important properties: completeness and closedness. Owing to the use of CGE in the EANT method, it has become possible to avoid the chromosome decoding phase.

The CGE scheme defines a genome as a linear sequence of genes capable of taking one of the three different forms (alleles): input, node or jumper. The input is the gene designating the input neuron. The node is the gene designating the neuron to which four parameters are related, namely: the weight, the current value of activation function, the GUID (Global Unique Identifier), and the number of input connections. The Jumper is the synaptic connection gene that stores references to the two nodes connected by the synaptic connection, and the GUID of the neuron to which the jumper is connected.

This method of genome representation can be interpreted as a linear program encoding a prefix tree-based program if one assumes that all the inputs of the neural network and all jumper connections are terminals, and the neurons are functions. A tree-based program can be stored in an array (linear genome) where the tree structure (topology of the neural network) is implicitly coded in the ordering of the elements of the array.

Pertaining to the advantages of the EANT method is as follows: a compact encoding of the genome, an absence of the decoding phase, and, as a consequence, a high operating speed. It should be noted that it is expedient to use this method in the systems having a constraint on the task solution time. A disadvantage of the EANT method is an absence of crossing-over operators and structural mutation operators for removal of neurons, and, as a consequence, restriction of the genetic search space.

ENS³ (Evolution of Neural Systems by Stochastic Synthesis) – the method inspired by and based on a biological theory of coevolution [15]. It is applied to neural networks of standard additive neurons with sigmoidal transfer functions and sets no constraints on the number of neurons and the topology of a network. The method develops neural network topology and parameters like bias terms and weights simultaneously.

The method is based on a behavior-oriented approach to neural systems. The algorithm originally was designed to study the appearance of complex dynamics in artificial sensorimotor systems for autonomous robots and software agents.

The method has advantages such as simplicity of implementation and good performance for small- and medium-sized neural networks. However, no crossing-over operator is implemented in the ENS³ method, which is considered to be a disadvantage, because when implemented properly, the crossing-over is capable of considerably reducing the evolution time and speeding up formation of an optimal individual. The ENS³ method also allows no modification of the pattern of neuron activation functions. Besides, fixed probabilities of mutation for neurons and connections reduce the search space: many effective configurations of neural networks cannot be formed due to low probability of changes in respective nodes of neural networks of the current population.

NEAT (Neuro-Evolution by Augmenting Topologies) – the method intended for optimizing weights and sequentially complexifying the structure of a neural network [16]. The initial population is generated from fully-connected neural networks consisting of input and output layers, where the number of neurons is predetermined. The genome structure in this method is based on the list of synapses. Each synapse stores the indices of two neurons (the signal source and receiver), the weight of the connection, an enable bit indicating whether the given synapse is active, and innovation number, which allows finding similar genes during crossing-over.

The method uses the direct encoding scheme and implements two mutation operators for separate modification of the weights and the structure; with the probability of mutation being fixed for each weight. The structural mutations increase the genome size owing to adding new genes, and add either the connection for two early unconnected neurons or the new neuron, whereas the existing connection is divided into two connections – the input and output of the new neuron. The replaceable connection is marked as inactive; the incoming connection weight is defined to be equal to one, and the outgoing connection weight is equated with the replaceable connection weight.

The crossing-over operator is based on biological concepts of homologous genes (alleles) and the synapsis process – alignment of homologous genes before crossing-over. The NEAT method uses innovation numbers – the historical markers associated with each gene for the purpose of tracking a chronology of changes to be made in that gene. The historical markers are recomputed in the following manner: whenever a new gene appears, the global innovation number is incremented and assigned to that gene. The gene of one of the individuals formed for crossing-over of a pair with the innovation number differing from all innovation numbers of genes of the other individual is called the disjoint gene.

The genes appearing in the given individual later than any of the other individual genes are called the excess genes. The genes with the same innovation numbers get aligned and form a genome for the next generation by mixing appropriate genes in a random manner or averaging the weights of connections. At the crossing-over stage, the probability of reactivation is specified for nonactivated genes. The NEAT method is effective with species within the population by way of separately computing the fitness of each species, which allows ensuring the genetic diversity. The population is divided into autonomous species by

introducing a metric measure into the space of genomes. For formation of separate species by clustering, the distance δ between chromosomes is introduced as a simple linear combination of the number of excess E and disjoint D genes, as well as the average weight differences of matching genes \bar{W} :

$$\delta = \frac{c_1 E + c_2 D}{N} + c_3 \bar{W}, \tag{1}$$

where N is the number of genes in the larger genome, which is the factor that normalizes for genome size, c_1, c_2, c_3 are the predetermine coefficients. The speciation based on the distance measure δ allows the problem of protecting the innovations to be solved.

Several modifications have been implemented for this method, the most successful of them are: rtNEAT (real-time NEAT), Phased Pruning, cgNEAT (Content-Generating NEAT), and HyperNEAT (Hypercube-based NEAT) – the method using the indirect encoding for adjustment of the parameters of a fully-connected neural network with the fixed structure.

The NEAT method has the advantages such as protecting topological innovations by historical markers and preserving a structural diversity in the population owing to specialization. These two approaches make it possible to solve the problems of premature convergence and unprotectedness of innovations. Nevertheless, the evolution by way of sequential complexification, which has been implemented in this method, is the cause of such disadvantages as search space restriction and high resource intensity.

The results of the comparative analysis of neuroevolutionary method are specified in Table 1 below.

Table 1 Comparative Characteristics of Neuroevolutionary Methods

Method	Sequence of modification of parameters and topology	Chromosome encoding technique	Evolution method
ENS ³	Parallel	Direct	Evolutionary algorithm
NEAT	Separate	Direct	Genetic algorithm
EANT	Separate	Hybrid	Evolutionary strategies
DXNN	Separate	Direct and indirect	Memetic algorithm
CE	Parallel	Indirect	Genetic programming

Based on the above analysis of methods, one can make the following conclusions: most methods fail to modify the activation function type and its parameters and, at this, impose constraints on the neural network structure; the evolution in many methods runs exceptionally in a way of complexifying (in some cases – sequentially simplifying) the structure of an individual; some methods take a supervised learning approach, which requires the availability of representative case-based samples and additional constraints on the neural network structure.

Therefore, none of the existing methods combines such properties as absence of constraints on the individual to be optimized, dynamic nature of evolution, and modification of the most of all the allowable parameters of a neural network.

In this connection, it seems actual to develop a novel neuroevolutionary method [17], which is free from the above-mentioned constrained and intended not only for adjusting the neuron weights and modifying the topology, but also for adjusting the threshold coefficients, type and parameters of activation functions. The method proposed in the article is distinguished by an ingenious collection of properties, the main of which are: structural adaptivity and low connectivity of individuals, dynamic nature of evolution, as well as possibility of hybridization. As distinct from most traditional neuroevolution

implementations, the method is free from such disadvantages as competition of representations, unprotectedness of innovations, problems of original size and topological innovations.

3. A NOVEL NEUROEVOLUTIONARY METHOD KHO FOR DECISION SUPPORT UNDER UNCERTAINTY

As a data structure for genetic encoding, an adjacency list has been chosen. The memory size taken by the list is linearly dependent on the number N of neurons and the number S synaptic connections and is expressed as the function of $O(N + S)$. A compact form of representation opens an opportunity to operate the neural network topology with an extra-large number of neurons. The adjacency list assigns a list of pairs in the form of $\{s_{ij}, w_{ij}\}$ to each indexed neuron i , where s_{ij} is the index of the neuron connected with j , w_{ij} is the connection weight, $i \in [1, N]$. For input neurons, this list is empty. For each neuron, the memory also stores the threshold value θ , type f_{type} and K of parameters r_0, \dots, r_K of activation function f . Each i -th element in the adjacency list is given as follows:

$$\{i, \{\{s_{i0}, w_{i0}\}, \dots, \{s_{iM}, w_{iM}\}\}, \theta, f_{\text{type}}, \{r_0, \dots, r_K\}, IN_i, OUT_i\}, \quad (2)$$

where IN_i is the minimum number of neural network nodes through which the input signal is to pass to reach the i -th neuron, OUT_i is the minimum number of nodes through which the signal generated by the i -th neuron is to pass to reach the output.

Storage and recalculation of the parameters IN_i and OUT_i , as well as indexing of the neural network nodes, encourage the solution of such problems as competition of representations and unprotectedness of innovations, which are typical for the direct encoding scheme: the new nodes added to the neural network by the mutation and crossing-over operators have larger indices than those formed in previous epochs of neuroevolution. The availability of information about the shortest path to the input and output nodes prevents the crossing of the neural network sections that carry different functional loads. Mutation in the neural network sections changed during previous epochs is less probable than that in the unevolved sections. In particular, the indexing of nodes reduces the risk of removal of new elements from the population, which solves the problem of unprotected innovations. It is the aforesaid universal representation that allows encoding of neural networks of any structure and size.

The proposed method utilizes two pools containing, respectively, input parameters and the variety of possible neuron activation functions. Based on the pool of input parameters constituting the majority of potential inputs of the neural network, the input vectors of individuals are formed in the evolution process. This pool is intended for optimizing the neural network efficiency, with the quality of network inference being maintained. The pool of activation functions contains the functions such as sigmoid function, Gaussian function, modified hypertangent, etc. The pool is required for adjusting the parameters of each neuron and, accordingly, increasing the accuracy of neural network inference. Along with the encoding method, the selection of genetic operators and the type of fitness function have a direct impact on the neuroevolution efficiency.

This method makes it possible to explore the search space in full, avoid local extremes at the genetic search stage, and efficiently use "good" solutions to be found, i.e. successively improve the results on the basis of intermediate solutions.

4. EXPERIMENTAL INVESTIGATION OF NEUROEVOLUTIONARY METHODS ON THE BASIS OF A SERIES OF BENCHMARK TASKS

As adaptive control tasks, three variants of tasks on the "inverted pendulum" theory and automatic control have been chosen. The inverted pendulum has a center of mass above its support point and is positioned at the end of a rigid pole whose support point is attached to a cart. At the initial moment of time, the pendulum is deflected through the selected angle from the equilibrium position. The task consists in setting the pendulum to the steady state by moving the cart with some force applied. In doing so, it is advised to avoid the ends of the track section. Three variants of the balancing system have been considered.

Classical task on inverted pendulum balancing. Given is an inverted pendulum with its cart moving within an interval of $-4 < x < 4$ m (Fig. 2). The cart original coordinate x_0 is preset randomly within an interval of $-2 < x_0 < 2$ m, with a cart initial velocity of $v_0 = 0$ m/s.

The initial angle φ_0 rad of pendulum deflection is set in a random manner, and the initial angular velocity of the pendulum is equal to zero: $\omega_0 = 0$ rad/s. The control signal is represented by the force F applied to the cart: $-10 < F < 10$ N.

The inverted pendulum balancing task is considered to be solved successfully, if the neural network managed to hold the pendulum for 30 minutes, not moving the cart out of the preset interval.

Task on double inverted pendulum balancing. The pendulum poles have a common support point on the cart moving along the straight line. The cart with two inverted pendulum poles with lengths l_1 and l_2 moves within an interval of $-4 < x < 4$ m. The cart original coordinate x_0 (m) is preset randomly within an interval of $-2 < x_0 < 2$ m, with a cart initial velocity of $v_0 = 0$ m/s. The initial angles φ_1^0 and φ_2^0 rad of deflection of both the pendulums are set in a random manner, and their initial angular velocities are equal to zero: $\omega_1^0 = \omega_2^0 = 0$ rad/s. The control signal is represented by the force F applied to the cart: $-10 < F < 10$ N.

The task on two inverted pendulums balancing is considered to be solved successfully, if the neural network managed to hold both the pendulums for 30 minutes, not moving the cart out of the preset interval.

Task on inverted pendulum balancing on plane. The cart with its attached inverted pendulum moves along the straight line, but in a two-dimensional space. Such a system has two degrees of freedom; the force F can be applied both along the X-axis and along the Y-axis and consists of two components: F_x and F_y . The system state is determined through eight parameters: the angles of pendulum deflection along axes X and Y (φ_x , φ_y), respective angular velocities of the pendulum (ω_x , ω_y), coordinates of the cart on the plane (x , y), and velocities of the cart along the axes (v_x , v_y).

The original coordinates (x_0, y_0) are preset randomly: $x_0 \in [-4, 4]$, $y_0 \in [-4, 4]$. The cart movement area is 20×20 m in size: $x \in [-10, 10]$, $y \in [-10, 10]$. The angular velocities of the pendulum are equal to zero: $\omega_{x0} = \omega_{y0} = 0$. The initial angles of deflections φ_{x0} and φ_{y0} are preset in a random manner. The control signal is represented by the force F applied to the cart: $-10 < F_x < 10$ N, $-10 < F_y < 10$ N.

The task on pendulum balancing on a plane is considered to be solved successfully, if the neural network managed to hold the pendulum for 30 minutes, not moving the cart out of the preset rectangular area.

Task on classification of unit hypercube points. To verify the efficiency of neuroevolutionary methods, the XOR function implementation task has been chosen for the construction of a classifier of linearly unseparable patterns, which is a particular case of the task on classification of unit hypercube points. Although the implementation of logical functions, in itself, by use of a neural network is considered trivial, it was used for testing for two reasons. First, testing for optimization of parameters: weights, type of activation functions, and threshold values. Implementation of the XOR function clearly demonstrates the accuracy of adjustment of weights and the optimality of choosing the threshold function as an activation functions. Second, testing for relevance of the neural network structure to be generated. For successful solution of the XOR task, a multilayer perceptron should have two inputs, a hidden layer of four neurons, and one output.

A single hidden layer should be formed in the neural network, which consists of two neurons with threshold activation functions. A learning set based on the XOR function truth table has been formed to solve this task by using neural networks to be generated by the neuroevolutionary methods.

Tasks on recovery of damaged data (noisy signals, damaged images). They are exponential, due to poor formalization, when testing the neuroevolutionary methods. As a task of such nature, the recovery of a damaged image not larger than $R \times C$ px in size, with X % pixels being unknown, has been chosen. The recovery process is carried out in the following way: an image containing X % damaged pixels is delivered to the input of the learned neural network. After that, a recovered pattern is formed at the output of the neural network. X % pixels in the network are substituted with weight values of relevant neurons. In process of the experiment, use was made of images with a color depth of 16 bits and with parameters of $R = 600$ and $C = 400$. Digital noise in the form of a mask of random-color and random-brightness pixels superimposed on the image is generated at $X \in [10; 80]$.

It is a good practice to evaluate the algorithm operation results with the use of the image recovery value ρ calculated by the following three parameters: a distance from the damaged image to the source one $d(I_{damaged}, I_{source})$, a distance from the recovered image to the source one $d(I_{recovered}, I_{source})$, and a distance from the damaged image to the recovered one $d(I_{damaged}, I_{recovered})$. For calculation of distance data, the Minkowskian metric is used:

$$d(U, V) = \max_{k,l} \sum_{j=1}^K \frac{|U_{kl}^j - V_{kl}^j|}{K} \quad (3)$$

where U_{kl}^j, V_{kl}^j are the color values of the j -th pixels of images U and V ; K the number of pixels. The images U and V are represented in the form of three color matrices corresponding to the RGB model. The image recovery value takes values on the segment $[0, 1]$ and is calculated by the following formula:

$$\rho = \frac{d(I_{damaged}, I_{source}) - d(I_{recovered}, I_{source})}{d(I_{damaged}, I_{recovered})} \quad (4)$$

Table 2 shows the results of solution of the above-mentioned benchmarks tasks by the basic neuroevolutionary methods and by the novel method (KHO) proposed in this article. To attain correct comparison of efficient of the methods, the task solution results have been averaged as per 100 starts. Evaluation has been made on the basis of the indirect measures of solution search efficiency by use of the neuroevolutionary methods, namely: the number of evolution epochs needed to solve the assigned tasks, and the number of failed neuroevolution starts whose results demonstrated that neither optimal nor close to optimal solution was formed [18].

Table 2 Results of Application of Neuroevolutionary Methods for Solution of Benchmark Tasks

Task	Number of epochs/failures				
Method	Inverted pendulum	Double inverted pendulum	Inverted pendulum on plane	XOR	Image recovery
CE	5/0	150/14	230/20	19/0	441/4
DXNN	3/0	614/28	802/12	9/0	235/0
EANT	7/0	975/14	1040/1	14/0	148/23
ENS ³	8/0	741/16	1021/5	15/1	360/0
NEAT	6/0	24/1	79/1	8/0	210/0
KHO	2/0	25/1	37/2	11/0	146/2

The obtained results attest to the fact that when an appropriate diversity of population individuals is achieved, the solution of trivial tasks (construction of logical XOR function and tasks on inverted pendulum balancing) can be formed in the first generations of neuroevolution, where the number of failed starts is equal to zero. The number of evolution epochs grows as the tasks become more complicated. Nevertheless, the results meeting the specified criteria were formed for all the tasks of the benchmarks series.

The proposed method KHO demonstrates the best or close to the best results and outperforms most previous methods as far as the number of epochs and the number of failed starts are concerned. Thus, the data obtained in process of the experimental investigation of efficiency of the method based on the neuroevolutionary benchmark tasks testifies that the method is efficient.

5. CONCLUSION AND PERSPECTIVES

One of the primary advantages of neuroevolutionary methods is their represented approach unified as maximum as possible to the solution of tasks under uncertainty. In particular, for complex tasks, the most of which are NP-complete, the neuroevolutionary methods demonstrate splendid results. This is confirmed by the experimental investigation of efficiency of neuroevolutionary methods based on most practice-oriented benchmark tasks.

In the course of the investigation, the results of solution of benchmarks tasks by using the novel method presented in this article and the analyzed CE, DXNN, EANT, ENS³ and NEAT methods, have been obtained. The proposed method KHO demonstrates the best results when solving the tasks based on inverted pendulum balancing and image recovery, or close to the best results for the tasks based on double inverted pendulum balancing, inverted pendulum balancing on a plane, and implementation of the logical XOR function. The analysis of the experiment results allows making the conclusion that it is expedient to use the method in practice.

It is accepted that the neuroevolutionary approach, in spite of its growing popularity in implementing the decision support system modules, is considered as an alternative trend in the decision making theory. The article shows that the neuroevolutionary methods demonstrate high quality values when solving the tasks in diverse domains, including the tasks on recovery of damaged data, control of dynamic objects, and classification of linearly inseparable patterns.

From a scientific perspective, discovering how the brain thinks and makes decision are a major undertaking in the history of humankind. Bioinformatics provides computational and experimental tools to study the biological patterns, structures, and functions. Cognitive informatics investigates the internal information processing mechanisms and process of life-

cognition [19]. How data was transcribed into information? How information was translated into knowledge? These are fundamental questions that require further investigations. Data mining is an active field that exams the process of extracting hidden patterns from data. Knowledge discovery is a growing field to exam the process of converting the information into knowledge [20]. Recent advances in experimentation such as patch clamp recording, voltage- and ion-specific dyes, and confocal microscopy are providing data to facilitate further theoretical development for addressing fundamental issues that range from the sub-cellular to cell-ensemble to whole-system levels. We must synthesize information and mechanisms across these different levels for thorough understanding from molecule to ecosystem. This is perhaps the fundamental challenge facing mathematical and theoretical biology. Models of neural interactions lead to many interesting mathematical questions for which appropriate tools must be developed. Typically, networks are modeled by (possibly stochastic) systems of differential equations. In some simplified limits, these become nonlinear integro-differential equations. The question now becomes one of proving or otherwise demonstrating that the simplified models have the desired behavior. Furthermore, one must characterize this behavior as parameters in the model vary (i.e., understand the bifurcations in the dynamics). Another important point that mathematicians must address is the extraction of the underlying geometric and analytic ideas from detailed biophysical models and simulations [21]. Both bioinformatics and cognitive informatics will play important roles in the decision-making theory.

ACKNOWLEDGEMENTS

The research was supported by grant of Plekhanov Russian University of Economics

REFERENCES

- [1] Afanasev, M., Dneprovskaya, N., Kliachin, M. and Demidko, D. (2018) Digital Transformation of the Knowledge Management Process. In E. Bolisani, E. Di Maria and E. Scarso (Eds.), Proceedings of the 19th European Conference on Knowledge Management – ECKM 2018 Vol.1, pp.1-8, Padova, Italy: Academic Conferences and Publishing International Limited.
- [2] Nefedov, Y., Pavlekovskaya, I. and Afanasev, M. (2018) Change in Knowledge Representation in the Digital Economy In E. Bolisani, E. Di Maria and E. Scarso (Eds.), Proceedings of the 19th European Conference on Knowledge Management – ECKM 2018 Vol.1, pp.589-594, Padova, Italy: Academic Conferences and Publishing International Limited.
- [3] Musatova Zh.B., Mkhitaryan S.V., Nevostruev P.Yu., Sidorchuk R.R., Komleva N.V. (2016) Smart-technologies in Public Transport and their Perception by the Youth Audience // Indian Journal of Science and Technology, Vol 9(42).
- [4] Dneprovskaya, N., Urintsov, A., Komleva, N. and Staroverova, O. (2018) Evaluation Indicators of Knowledge Management in the State Service. In E. Bolisani, E. Di Maria and E. Scarso (Eds.), Proceedings of the 19th European Conference on Knowledge Management – ECKM 2018 Vol.1, pp.182-189, Padova, Italy: Academic Conferences and Publishing International Limited. 6-7 September 2018.
- [5] Komleva, N., Dneprovskaya, N. and Vnukova, T. (2018). Content Evaluation in Knowledge Management Systems. In E. Bolisani, E. Di Maria and E. Scarso (Eds.), Proceedings of the 19th European Conference on Knowledge Management – ECKM 2018 Vol.1, pp.399-406, Padova, Italy: Academic Conferences and Publishing International Limited. 6-7 September 2018.

- [6] Natalia V. Dneprovskaya, Nina V. Komleva, Arkadiy I. Urintsov. (2018). The Knowledge Management Approach to Digitalization of Smart Education. Proceedings of the Second International Conference of Artificial Intelligence, Medical Engineering, Education –AIMEE 2018, 6-8 October 2018, pp.600-609.
- [7] Irina Pavlekovskaya, Arkadiy Urintsov, Olga Staroverova and Yuriy Nefedov (2018). The Impact of Digital Transformation of the Russian Economy on Knowledge Management Processes. Proceedings of the 19th European Conference on Knowledge Management – ECKM 2018 Vol.2, pp.677-684, Padova, Italy: Academic Conferences and Publishing International Limited. 6-7 September 2018.
- [8] S. Haikin, (2006), Neural Networks: A Comprehensive Foundation, Second Edition / Simon Haikin, translated from English by N.N. Kussul. – Moscow: Publishing House "Williams", 2006. – p. 1104.
- [9] Tikhomirova, N., Komleva, N. (2010). Development of business models of open education based on new intellect technologies. The EADTU's annual conference 2010, Zermatt Switzerland, 27-29 September 2010, Proceedings 'Strategies and Business Models for Lifelong Learning/ Networking Conference', pp.400-406.
- [10] Komleva, N., Danchenok, L. and Gulaya, T. (2010) 'Innovative information environment for enriching education quality', Int. J. Foresight and Innovation Policy, Vol. 6, No. 4, pp.248–257.
- [11] A.P. Goriashko, E.V. Nikulchev, (2011), Robust Optimization as a New Approach to Solution of Tasks under Uncertainty // Bulletin (Izvestiya) of Higher Education Institutes. Problems of Polygraphic and Publishing Industry, No. 1. – pp. 77–86.
- [12] D. Whitley, F. Gruau, L. Pyeatt, (1995), Cellular encoding applied to neurocontrol // San Francisco: Sixth International Conference on Genetic Algorithms, pp. 460–469.
- [13] G.I. Sher, (2011), Evolving Chart Pattern Sensitive Neural Network Based Forex Trading Agents // Neural and Evolutionary Computing, Computational Engineering, Finance, and Science, pp. 47–68.
- [14] G. Sommer, Y. Kassahun, (2005), Efficient reinforcement learning through evolutionary acquisition of neural topologies, 13th European Symposium on Artificial Neural Networks, pp. 259–266.
- [15] F. Pasemann, U. Dieckmann, (1997), Balancing rotators with evolved neurocontroller. – Leipzig: Max–Planck–Institute for Mathematic in Sciences, pp. 36–43.
- [16] K. Stanley, R.O. Miikkulainen, (2002), Evolving neural networks through augmenting topologies // Evolutionary Computation, Vol. 10, No. 2, pp. 99–127.
- [17] Komleva N, Khlopkova O (2013) Handling content in the information environment based on neuro-fuzzy model of decision-making. Economics, Statistics, Computer Science. Vestnik UMO. Scientific and practical journal, No. 5, pp.188-192 (in Russian).
- [18] G.I. Sher, (2013), Handbook of neuroevolution through Erlang, N.Y.: Springer, p. 851.
- [19] He, M. (2008), On bioinformatics and cognitive informatics: emerging pattern, dissipative structure, and evolving cognition. In Y. Wang and W Kinsner (Eds.), Proceedings of the 8th IEEE International Conference on Cognitive Informatics (ICCI'08), Stanford University, Aug. 14-16, pp. 40-49.
- [20] He, M., (2009), The evolution and future trends of technology and information sciences, 2009 WASE International Conference in Information Engineering, Vol. 1, pp 11-15.
- [21] He, M., Petoukhov, S.V., (2011), Mathematics of Bioinformatics: Theory, Practice, and Applications, John Wiley & Sons.

AUTHORS' PROFILES



Komleva Nina V., Ph.D in Mathematics and Tools of Economics, Associate Professor of Information Systems Management and Programming Department of Plekhanov Russian University of Economics, Moscow, Russia, experience in the development of e-learning and Knowledge Management in Russian universities for over 20 years. The expert of World Skills Russia within the higher education area. The main scientific researchers are in models and tools of innovative development of education in an open information environment, Neural Networks and Artificial Intelligence.



Khlopkova Olga A., Ph.D., In 2013 graduated from the Moscow state university of economics, statistics and informatics, Moscow, Russia, From 2013 until 2016 she worked under the guidance of Prof. Komleva Nina V. on the dissertation focused on neural network training to support decision making. In the July 2016 received Ph.D. in Informatics.



Matthew He, Ph.D., Assistant Dean/Full Professor of the Halmos College of Natural Sciences and Oceanography of Nova Southeastern University (NSU) in Florida. From 2004 to 2015, Prof. He served as the Director of the Division of Math, Science, and Technology of Farquhar College of Arts and Sciences of the university. He has been a full professor and grand Ph.D. of the World Information Distributed University since 2004, as well as an academician of the European Academy of Informatization. He was a research associate at the Department of Mathematics, Eidgenossische Technische Hochschule, Zurich, Switzerland, and the Department of Mathematics and Theoretical Physics, Cambridge University, Cambridge, England. He was also a visiting professor at the National Key Research Lab of Computational Mathematics of the Chinese Academy of Science and the University of Rome, Italy.

Dr. Matthew He has authored/edited 10 books and published over 100 research papers in the areas of bioinformatics, computational biology and mathematics, non-rigid motion analysis, and information theory. Dr. Matthew He serves as a member of board directors of many international associations, keynote speakers of many international conferences in math, information sciences, bioinformatics, biomedical sciences and bioengineering, and is actively involved in promoting science, education, and technology.