I.J. Intelligent Systems and Applications, **2024, 4, 1-21**
Published Online on August 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2024.04.01

Polymorphic Radial Basis Functions Neural Network

Serhii Vladov
Scientific Work Organization and Gender Issues Department, Kremenchuk Flight College of Kharkiv National University of Internal Affairs, Kremenchuk, 39605, Ukraine
E-mail: ser26101968@gmail.com
ORCID iD: https://orcid.org/0000-0001-8009-5254

Ruslan Yakovliev
Air Navigation Department, Kremenchuk Flight College of Kharkiv National University of Internal Affairs, Kremenchuk, 39605, Ukraine
E-mail: director.klk.hnuvs@gmail.com
ORCID iD: https://orcid.org/0000-0002-3788-2583

Victoria Vysotska
Information Systems and Networks Department, Lviv Polytechnic National University, Lviv, 79013, Ukraine
E-mail: Victoria.A.Vysotska@lpnu.ua
ORCID iD: https://orcid.org/0000-0001-6417-3689

Dmytro Uhryn*
Department of Computer Science of the Yuriy Fedkovych Chernivtsi National University, Chernivtsi, 58012, Ukraine
E-mail: d.ugryn@chnu.edu.ua
ORCID iD: https://orcid.org/ 0000-0003-4858-4511
*Corresponding author

Artem Karachevtsev
Department of Computer Science of the Yuriy Fedkovych Chernivtsi National University, Chernivtsi, 58012, Ukraine
E-mail: a.karachevtsev@chnu.edu.ua
ORCID iD: https://orcid.org/0009-0000-6226-6822

Received: 12 October 2023; Revised: 14 January 2024; Accepted: 07 March 2024; Published: 08 August 2024

Abstract: The work is devoted to the development of the radial basis functions (RBF networks) neural network new architecture – a polymorphic RBF network in which the one-dimensional radial basis functions (RBFs) in the hidden layer instead, multidimensional RBFs are used, which makes it possible to better approximate complex functions that depend on several independent variables. Moreover, in its second layer, the summing the RBF outputs one by one from each group instead, multiplication is used, which allows the polymorphic RBF network to better identify relations between independent variables. Based on the training classical RBF networks evolutionary algorithm, the polymorphic RBF network training algorithm was created, which, through the initializing weight coefficients methods use taking into account the tasks structure and preliminary values, using the mutations tournament selection, adding additional criteria to the fitness function to take into account stability and speed training a polymorphic RBF network, as well as using an evolutionary mutation strategy, allowed us to obtain the lowest errors in training and testing a polymorphic RBF network compared to known RBF network architectures. The created polymorphic RBF network practical application possibility is demonstrated experimentally using the helicopters turboshaft engines (using the example, the TV3-117 turboshaft engine) operating process parameters optimizing solving task using a multicriteria optimization algorithm. The optimal Pareto front was obtained, which made it possible to obtain the engine operation three additional modes: maximum reduction of specific fuel consumption at the total pressure in the compressor increase degree increased value by 5.0 %, specific fuel consumption minimization at the total pressure in the compressor increase degree reduced value by 1.0 %, the total pressure in the compressor increases degree optimal value with a slight increase in specific fuel consumption by 10.5 %. Future research prospects include adapting the developed methods and models into the general concept for monitoring and controlling helicopter turboshaft engines during flight operations. This concept is implemented in the neural network expert system and the on-board automatic control system.

Index Terms: Polymorphic RBF Network, Training, Multi-criteria Optimization, Pareto Front, Radial Basis Functions.

1. Introduction

Radial Basis Function (RBF) neural networks are a powerful tool for solving a tasks wide range due to their simple architecture and high representational power [1, 2]. The RBF networks performance is highly dependent on the data clustering efficiency, which determines the basis functions centers and dispersions. A crucial architectural parameter is the basis functions number, which sets the approximation units number limit. Correctly selecting this number balances the network's generalization ability and its accuracy in fitting specific training data [3–7].

The RBF network with a dynamic hidden layer structure first widely recognized model, enabling the basis functions' number automated selection, was the Platt model [8]. This model was subsequently improved in various works [9, 10]. Research on automating RBF network architecture selection focuses on the training algorithms local modifications [11, 12], the bionic models' application [13, 14], and the basis functions number optimizing [15, 16]. The latter approach shows significant potential, leading to increased interest in using distributed intelligent systems for optimizing neural network architectures [17, 18]. An alternative method involves density clustering to determine the basis functions optimal number and establish their key characteristics [19, 20].

However, most proposed solutions require complete processing, making them inefficient for systems with dynamically changing data, such as control systems, compared to specialized neuroarchitectures like Jordan [21], Elman [22], or multilayer perceptron recurrent networks [23, 24].

A radial-basis functions key property is the monotonic and symmetrical their responses decrease relative to the given center. In the classical RBF network structure, there are only neurons two layers: the first hidden layer contains radial elements that process the input vector by evaluating the input signals proximity to the center coordinates, and the second layer calculates the first layer's outputs linear combinations. Properly configured RBF networks have a simple structure, low calculation error, and high training rate [25, 26].

A novel RBF network structure was proposed in [27], where independent variable each signal is fed into a separate hidden layer with one-dimensional radial elements. The radial basis functions number differs in each group. The second layer sums the radical elements outputs in all possible combinations and squares them. The output layer neurons weight coefficients are calculated similarly to traditional RBF networks.

This architecture has several disadvantages: increased computational complexity, more parameters, potential overfitting, and requiring a training data large amount. Additionally, radial basis functions complex distribution configurations complicate network design and optimization.

Another architecture, proposed in [28], integrates one or more additional layers similar to a perceptron (hybrid RBF network). Although it requires more calculations, it provides lower training errors compared to classical RBF networks. However, the testing errors of both hybrid and traditional RBF networks are similar, indicating the need for additional tuning and larger training samples to fully realize the hybrid architecture potential.

The RBF networks another disadvantage is their high tendency to overfit training data, reducing generalization ability on new, unseen data [29]. This issue is particularly problematic with limited or noisy training data, leading to the model universality loss and predictive ability on real data. Addressing this problem may involve modifying RBF network architecture or employing more sophisticated training algorithms and optimization methods for the basis functions number and distribution.

In summary, while the traditional RBF network structure is simple and advantageous, it has limitations. RBF networks can train directly from data, offer low-cost computation, and adapt to changes, but they are vulnerable to insufficient training data, which can cause overfitting. Additionally, selecting the appropriate number and placement of centers is challenging, especially for high-dimensional data, and training can be computationally expensive, limiting their application in large-scale problems.

The aim of the work is to create a polymorphic RBF network – a modified architecture of the traditional RBF network with a dynamic hidden layer structure to improve its generalization ability and resistance to retraining. A polymorphic RBF network is expected to have the following advantages over traditional RBF networks:

- better generalization ability – a polymorphic RBF network will generalize better on new data, making it more resistant to overtraining,
- increased robustness to noise – the polymorphic RBF network will be more robust to noise in the training data,
- reduction of computational complexity – a polymorphic RBF network can have a lower computational complexity of training, due to the dynamic structure of the hidden layer.

The work scientific novelty consists in the RBF network new architecture development – a polymorphic RBF network with the hidden layer dynamic structure, which allows to improve its generalization ability and resistance to retraining.

2. Material and Methods

In this work, a polymorphic RBF network is proposed as the traditional RBF network modification, which has the following features:

- multidimensional radial basis functions – one-dimensional radial basis functions instead, multidimensional radial basis functions are used in the hidden layer, which allows for better approximation of complex functions that depend on several independent variables,
- the radial elements outputs multiplication – in the second layer, instead the RBF outputs summing, one from each group, multiplication is used, which allows the RBF network to better identify relationships between independent variables,
- dynamic structure of the hidden layer – the number and placement of RBF centres are determined automatically during network training, which makes it more resistant to retraining.

The proposed polymorphic RBF network corresponds to the following statement (proposition): if $f(x_1, x_2, ..., x_n)$ – some continuous function that depends on n independent variables, and $\varepsilon > 0$ is the arbitrary number, then there exists a polymorphic RBF network with a sufficient number of multidimensional radial basis functions in the hidden layer and an summation multiplication instead optimal use in the second layer, which is able to approximate f with an accuracy no worse than ε. For any $\varepsilon > 0$, it is possible to construct a sufficiently large polymorphic RBF network with multidimensional radial basis functions in the hidden layer. Given a functions sufficient number in each group and this network parameters proper tuning, we can approximate the function f on any input data finite set with an accuracy that is less than ε. This can be achieved according to the approximation theory. Replacing the radial elements outputs summation, one from each group, with multiplication allows the network to detect relations between independent variables with better accuracy. With an appropriate parameter setting and a radial functions sufficient number in the second layer, a polymorphic RBF network can approximate the function f with any accuracy.

So, the proposed new structure of the RBF network (fig. 1) at the input has a vector of input parameters $x = (x_1, x_2, \ldots, x_n)$. The hidden layer contains m neurons, each of which is a multidimensional radial basis function, and each hidden layer neuron weights are calculated as [30–33]:

$$w_{ij} = \varphi\left(\|x-c_i\|\right) = e^{-\frac{\left(\|x-c_j\|\right)^2}{2\cdot\sigma_i^2}}, \tag{1}$$

where $\|x-c_i\| = \sqrt{\sum_{j=1}^{N}\left(x_j - c_{ij}\right)^2}$ is the Euclidean distance between the input vector $x = (x_1, x_2, \ldots, x_n)$ and the i-th ($i = 1\ldots L$) neuron $c_i = (c_{i1}\ldots c_{iN})$ center; L is the hidden layer neurons number; N is the input layer neurons number c_i and σ_i are radial basis function parameters for the i-th neuron. The output layer neuron signal is determined by the hidden layer neurons outputs weighted summation $f_k = \sum_{i=1}^{L} w_i \cdot y_i$, where w_i is the connection weight from the hidden layer i-th neuron to the output layer neuron. Introducing the notation $\mathbf{z} = (z_1\ldots z_p)^T$ is the expected function values vector (with p being the training samples number), w = $(w_1\ldots w_L)^T$ as the weights vector, **G** is the radial matrix, which, according to [34], has appearance:

$$G = \begin{pmatrix} \varphi\|x_1-c_1\| & \varphi\|x_1-c_2\| & \ldots & \varphi\|x_1-c_L\| \\ \varphi\|x_2-c_1\| & \varphi\|x_2-c_2\| & \ldots & \varphi\|x_2-c_L\| \\ \ldots & \ldots & \ldots & \ldots \\ \varphi\|x_p-c_1\| & \varphi\|x_p-c_2\| & \ldots & \varphi\|x_p-c_L\| \end{pmatrix}, \tag{2}$$

The weights vector is calculated as:

$$\mathbf{w} = \mathbf{G}^{+} \cdot \mathbf{z}, \tag{3}$$

where $\mathbf{G^{+} = (G^{T}G^{-1})G^{T}}$ is the rectangular matrix **G** pseudo-inversion.

Therefore, the hidden layer i-th neuron can be fully described by a (N + 2) real numbers vector, comprising $c_i = (c_{i1}\ldots c_{iN})$, σ_i, and w_i. Thus, describing the entire network requires a matrix **R** of dimensions $L \times (N + 2)$. According to [34], it is recommended to a self-adaptive weight adjustment method use, so the same size matrix $\boldsymbol{\upsilon}$, containing variations (evolutionary algorithm strategic parameters), is also included in the neuron description.

The output layer contains k neurons, and the output layer each neuron output is calculated as:

$$y_k = w_k \cdot \sum_{p=1}^{m} \left(w_p \cdot f_p\left(x\right) \right)^2, \tag{4}$$

where x – vector of input data, f_p – multidimensional radial basis function of the i-th neuron of the hidden layer, s_r – center of the p-th radial basis function, σ – width of the Gaussian function, w_k – weight of the k-th neuron of the output layer.

The proposed polymorphic RBF network (fig. 1) will improve its ability to approximate complex functions that depend on several independent variables. This is because the multivariate radial basis functions use allows the network to better account for the data spatial structure, and the multiplication use in the second layer allows the network to better detect the relations between independent variables.

In [35], it was noted that an RBF network cannot achieve a steady state during training if there are elements with similar center coordinates c_{ij} and radial function widths σ_i. This situation is influenced by the elements chosen number and their initial parameters. The degradation in training quality arises because the gradient algorithm assumes that the any RBF network output value at each point is primarily influenced by a single element. When multiple elements are present in one area, adjusting their parameters using the gradient algorithm does not always reduce the training error.

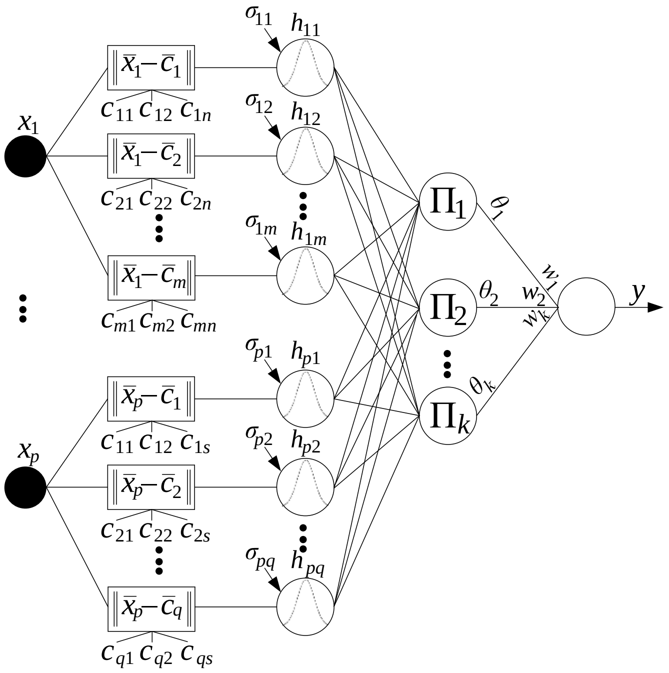


Fig.1. The proposed polymorphic RBF network (author's research)

To identify situations where the certain elements parameters become similar, [35] introduced the elements mutual intersection coefficient concept. To calculate this coefficient for an element in any RBF network, one must find the second element whose center is nearest to the element center being analyzed. The mutual intersection coefficient is defined as the first element value sum at the second element center and the second element value at the first element center:

$$\rho_i = e^{-\frac{\sum_{j=1}^{n}\left(c_{ij} - c_{dj}\right)^2}{2\cdot\sigma_i^2}} + e^{-\frac{\sum_{j=1}^{n}\left(c_{ij} - c_{dj}\right)^2}{2\cdot\sigma_d^2}}, \tag{5}$$

where i is the element index for which the mutual intersection coefficient is calculated, d is the element index whose center is the i-th element center closest. The proximity is determined using the following expression:

$$d = \arg\min_{k} \sqrt{\sum_{j=1}^{n}\left(c_{ij} - c_{kj}\right)^2}. \tag{6}$$

According to [28, 29], the mutual intersection coefficient ranges between 0 and 2, reaching its maximum when the centers of the analyzed elements coincide. To ensure stability, this coefficient should be capped at 1.95. In [35], a training RBF networks evolutionary algorithm is proposed, with its main limitations summarized in Table 1.

Table 1. Disadvantages of the evolutionary algorithm for training the RBF network (author's research based on [35])

Disadvantage	Description
Initial initialization of parameters	Parameters random initialization from the interval (1,0; 1,0) can lead to the fact that some parameters will be too close to this interval limits, which can limit the algorithm maneuverability.
Adaptability function	The analytical expression for calculating fitness takes into account only the network actual output deviation from the expected one. This may not be adaptability sufficient measure for training in order for the network to function harmoniously under different conditions.
Mechanism of selection for mutation	Individuals' mutation is based on their rank, but this approach can result in fewer fit individuals being selected for mutation, while more fit ones may remain unchanged.
Mutation method	Gaussian mutation can cause changes in the activation function parameters to become too large or too small, which can make it difficult for the algorithm to converge.
"Greedy" algorithm	Employing a "greedy" algorithm, where the neuron removal is attempted before adding one, can result in missing optimal solutions. This is because neurons might be removed even when there is potential for improvement.
Elitism	Using the elitism principle can lead to insufficient diversity in the population and new solutions insufficient development.
Unsuccessful mutations	In unsuccessful mutation case, the individual is copied without changes. This can lead to bad decisions remaining in the population without getting enough chance to improve.

The algorithm possible modifications one may be to improve the parameter initialization process, using more efficient strategies, such as generating weight coefficients methods according to the task specific characteristics. Also, you can consider optimizing the selection mechanism for mutation and improving the fitness function:

1. Improved initialization. Randomly initializing instead from the interval (1,0; 1,0), the weights initializing more intelligent methods use, such as methods that take into account the task structure or use prior knowledge:

- "No" initialization is the initialization is designed specifically for neural networks and provides a training more efficient start, especially for ReLU (Rectified Linear Unit) type activation functions. The weights are initialized according to $w=\{\text{random}\}\cdot\sqrt{\frac{2}{N}}$, where N is the previous layer size, {random} indicates that the weights that multiply the input x are initialized to random values,
- Xavier/Glorot initialization is the layer dispersion preservation input and output when multiplying by weighting factors. The weights are initialized according to $w=\{\text{random}\}\cdot\sqrt{\frac{2}{N+M}}$, where M is the current layer size.

Using prior knowledge – if there is prior knowledge of optimal weights for similar tasks or data, then this knowledge can be used to initialize the weights. For example, if there is a pre-trained model on similar data, the weights can be initialized from this model.

As an example, in dynamic system neurocontrol with a reference model (fig. 2) [36, 37], the RBF network weight coefficients values are calculated according to the gradient descent method [38, 39]:

$$\Delta w_j(k) = -\eta\cdot\frac{\partial E}{\partial w_{ij}} = \eta\cdot e\cdot c(k)\cdot\frac{\partial y(k)}{\partial u(k)}\cdot h_j\cdot w_j(k) = w_j(k-1)+\Delta w_j(k)+\alpha\cdot\Delta w_j(k), \tag{7}$$

where $\eta\in[0,1]$, $\alpha\in[0,1]$ is the training speed coefficients, $u(k)=w_1\cdot h_1+w_2\cdot h_2+\ldots+w_m\cdot h_m$ is the neuroregulator output, $y(k)$ – system output.

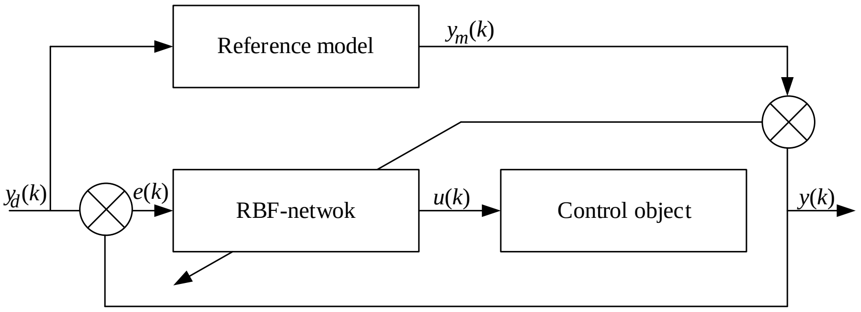


Fig.2. The adaptive neural network control system structure with a reference model [35]

The radial basis functions parameters are determined as follows:

$$\Delta b_j\left(k\right) = -\eta \cdot \frac{\partial E}{\partial w_{ij}} = \eta \cdot e \cdot c\left(k\right) \cdot \frac{\partial y\left(k\right)}{\partial u\left(k\right)} \cdot \frac{\partial u\left(k\right)}{\partial b_j} = \eta \cdot e \cdot c\left(k\right) \cdot \frac{\partial y\left(k\right)}{\partial u\left(k\right)} \cdot w_j \cdot h_j \cdot \frac{\left\| x - c_{ij} \right\|^2}{b_j^3}, \tag{8}$$

$$b_j\left(k\right) = b_j\left(k-1\right) + \eta \cdot \Delta b_j\left(k\right) + \alpha \cdot \left(b_j\left(k-1\right) - b_j\left(k-2\right)\right), \tag{9}$$

$$\Delta c_{ij}\left(k\right) = -\eta \cdot \frac{\partial E}{\partial c_{ij}} = \eta \cdot e \cdot c\left(k\right) \cdot \frac{\partial y\left(k\right)}{\partial u\left(k\right)} \cdot \frac{\partial u\left(k\right)}{\partial c_{ij}} = \eta \cdot e \cdot c\left(k\right) \cdot \frac{\partial y\left(k\right)}{\partial u\left(k\right)} \cdot w_j \cdot h_j \cdot \frac{x - c_{ij}}{b_j^2}, \tag{10}$$

$$c_{ij}\left(k\right) = c_{ij}\left(k-1\right) + \eta \cdot \Delta c_{ij}\left(k\right) + \alpha \cdot \left(c_{ij}\left(k-1\right) - c_{ij}\left(k-2\right)\right). \tag{11}$$

2. Improved selection for mutation. The selecting individuals instead for rank mutation, use more efficient selection strategies, such as tournament selection, where several random individuals are selected for comparison. Let a N individuals' population be given. To hold a tournament, k individuals i_1, i_2, …, i_k (where k is the tournament size, usually use the value 2 or 3) are chosen randomly from the population. After that, these individuals are compared according to a certain metric $f(i_1)$, $f(i_2)$, …, $f(i_k)$ (for example, the fitness function value [40, 41]). The individual for mutation M is selected according to the expression:

$$M = \arg\max\left(f\left(i_1\right), f\left(i_2\right), ..., f\left(i_k\right)\right). \tag{12}$$

The individual with the best metric value is considered the winner of the tournament and is selected for mutation. Therefore, tournament selection allows selection of an individual for mutation taking into account its competitiveness in comparison with other randomly selected individuals from the population.

3. Advanced adaptability function. Additional criteria are added to the fitness function to consider not only the deviation of the original values, but also neural network other characteristics, such as robustness, training rate, etc. The population all individual's fitness is computed as [27] $e_m = \frac{1}{T} \cdot \sqrt{\sum_{i=1}^{T} \left(y\left(t\right) - z\left(t\right)\right)^2}$, where T is the training dataset samples number; $Y(t)$ and $Z(t)$ selection individual mechanism is based on its rank, which compares the expected and the network output actual values. Arrange K individuals in descending order based on function e_m and assign them numbers 0, 1, ..., (K – 1). The individual numbered (K – j) may then be chosen for mutation with a probability $p\left(K - j\right) = j \cdot \left(\sum_{k-1}^{K} k\right)^{-1}$. Prior to initiating mutation, an integer n is randomly chosen from the interval (1, L), indicating the neuron number to undergo the mutation operation.

4. Mutation improved methods. More complex mutation methods are used, allowing more flexibility in changing parameters, for example, mutation using an evolutionary strategy, which involves the following steps:

- parent selection is the typically, parent selection can be done, for example, using tournament selection, where multiple agents are randomly selected for comparison, and the one with the highest fitness is selected,
- mutations generation for each selected parent, its mutants are created. It is assumed that x is the father's parameters vector, and $N(0, \sigma^2)$ is the random vector with a normal distribution with a mean zero value and σ^2 variance. Then the mutation looks like: $x_{new} = x + \sigma N(0, 1)$,
- mutations application is the random changes are applied to each parameter. For example, you can use the Gaussian mutation $x_{new} = x + \sigma N(0, 1)$,
- the mutant's fitness estimating for each new agent, its fitness is calculated, for example, using the fitness function $fitness = f(x_{new})$,
- the new generation selection those agents with the highest fitness values are selected. This can be a top gents' selection from a sorted list based on their fitness function,
- iteration – these steps are repeated for several generations, with the agents' genetic characteristics improving aim.

5. Dynamic strategies. Dynamic strategies are implemented to adapt the mutation probability and other parameters in the evolution process, which can improve the algorithm convergence. For example, let $p(t)$ be the mutation probability at time t, and let $f(x)$ be the fitness function value for individual x. One of the possible strategies is the change of $p(t)$ depending on the fitness function values dynamics. An illustrative example is the mutation probability adaptive control using recursive rules [42, 43]:

$$p(t+1) = p(t) + \alpha \cdot \Delta f(t), \tag{13}$$

where $p(t)$ is the mutation probability at time t, $p(t + 1)$ is the mutation probability at time $t + 1$, α is the change coefficient, $\Delta f(t)$ is the fitness function value change between times t and $t + 1$. The sign $\Delta f(t)$ indicates whether the fitness function value has increased (positive value) or decreased (negative value).

3. Experiment

It is known [44, 45] that RBF networks are a powerful tool in optimization problems, especially in cases where the input data have a complex unstructured form. Their main advantages are the ability to train non-linear dependencies, high learning speed and relative simplicity of implementation. The use of RBF networks in optimization problems makes it possible to efficiently find optimal solutions even in complex, multidimensional spaces, making them an important tool in many fields, including aviation, engineering, cyber security, and data science.

Based on the aforementioned discussion, the research conducted a computational experiment using the proposed polymorphic RBF network, aimed at the helicopter turboshaft engines (TE) parameters optimizing [46]. The helicopter TE operational parameters, crucial for the thermogas-dynamic equations governing the engine, were selected for optimization [47], with some parameters predicted rather than optimized in specific cases [48]. The primary focus in optimizing helicopter TE operation is on key parameters such as gas temperature in front of the compressor turbine or the increase in total compressor pressure to achieve optimal system performance [46]. Given that the efficiency criteria exhibit quadratic-like dependencies on the operational parameters, employing a second-order elliptic paraboloid model is recommended for their approximation. The least squares method is suitable for solving this task due to its simplicity and reliable function approximation capabilities. Robust methods for evaluating experimental results are also considered to mitigate significant errors [46]. The helicopter TE operation generalized regression model, wherein parameters are optimized, is formulated as follows according to [46]:

$$y = ax_1^2 + bx_2^2 + cx_1x_2 + dx_1 + ex_2 + j, \tag{14}$$

where x_1 denotes an independent variable correlating to the pressure level increase in the compressor, and x_2 represents an independent variable correlating to the gas temperature in front of the compressor turbine. Coefficients a, b, c, d, e, f in this model is estimated using the least squares method.

To find the function y partial derivatives and determine its minimum or maximum along with the corresponding values x_1 and x_2 according to the equations system [46]:

$$\begin{cases} y'_{x_1} = 2ax_1 + cx_2 + d = 0, \\ y'_{x_2} = 2bx_2 + cx_2 + e = 0. \end{cases} \tag{15}$$

For optimizing single-parameter tasks, this approach enables the functions utilization $y = f\left(T_G^*\right)$ or $y = f\left(\pi_C^*\right)$ (fig.3).

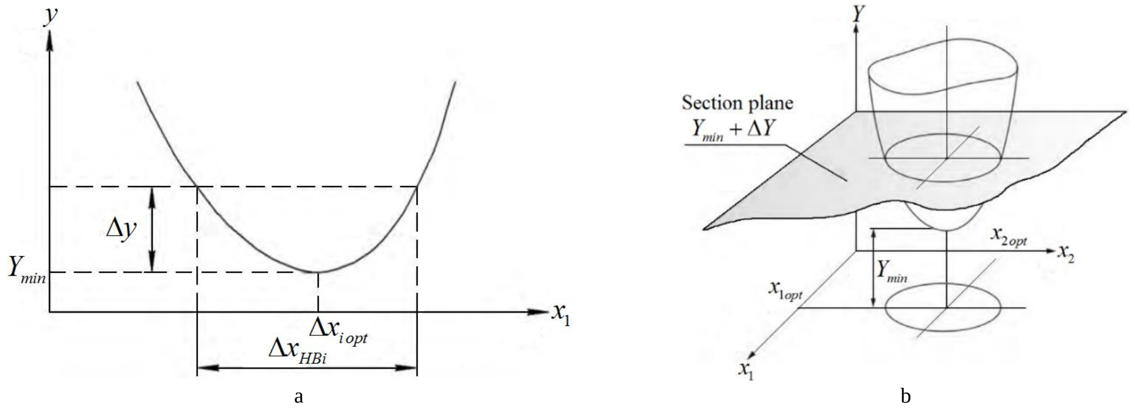


Fig.3. Diagram of the most favorable parameters range formation for one-parameter tasks (a) and the regions of rational parameter values for two-parameter tasks (b) [46]

Finding an objective function combined solution and a plane, offset from the extrema by ΔY, results in a closed curve resembling an ellipse on the $x_1 - x_2$ ($\pi_C^* - T_G^*$) plane for each criterion function Y_i. These lines actually define the workflow parameters rational values areas boundaries.

Based on [35, 46], the task involving the multiple criteria simultaneously minimization with m independent variables, n goals, p inequality constraints and q equality constraints looks like “the $f(x)$ minimization under the condition $g(x) > 0$, $h(x) = 0$”, where $x = (x_1 \ldots x_m) \in X$ is the solutions vector, X is the parameters space, $f(x)^T = [f_1(x)\ldots f_n(x)]$ are the aims, $g(x)^T = [g_1(x)\ldots g_p(x)]$ are constraints in the inequalities form, $h(x)^T = [h_1(x)\ldots h_q(x)]$ are constraints in the equalities form. The solution vector $a \in X$ dominates the vector $b \in X$ if the following condition is met

$$\forall i \in \{1,...,n\} : f_i(a) \le f_i(b) \wedge \exists j \in \{1,...,n\} : f_j(a) < f_j(b). \tag{16}$$

A vector a is non-dominated on the set $X' \subseteq X$, if X' there is no vector dominating over a. The solutions X' set for which the condition [46, 49] is fulfilled:

$$\forall a' \in X' : \neg\exists a \in X : a \prec a' \wedge \left\|a - a'\right\| < \varepsilon \wedge \left\|f(a) - f(a')\right\| < \delta. \tag{17}$$

where $\|...\|$ is the distance metric, when $\varepsilon > 0$, $\delta > 0$ is the local Pareto-optimal set, X' is the global Pareto-optimal set if $\forall a' \in X' : \neg\exists a \in X : a \prec a'$ [35, 46].

Hence, the multicriteria optimization aim is to discover the solutions global set that are Pareto-optimal. During helicopter TE flight operations, this set is presented to the helicopter crew commander, who selects a regulatory policy to determine subsequent flight options. References [35, 46] advocate employing multi-criteria optimization methods, integrating heuristic approaches like evolutionary and genetic algorithms (see table 2, fig. 4).

Table 2. Multi-criteria optimization method main stages [34, 46]

Step	Description
1	An initial training sample x_s of limited size $s \in X$ is created using an experimental design method, such as one outlined in [50]. The vectors representing the objective functions $f(x_s)$ as well as the constraints $g(x_s)$ and $h(x_s)$, are then computed for all acquired points.
2	Using the training sample x_s and the $f(x_s)$, $g(x_s)$ and $h(x_s)$ corresponding values, approximate models of $f(x)$, $g(x)$ and $h(x)$ for all researched relationships are constructed.
3	Using the derived approximate models for $f(x)$, $g(x)$ and $h(x)$ the NSGA-II algorithm identifies the x_{opt} vector, which defines the Pareto-optimal solutions set for the multi-criteria optimization task.
4	The functions $f(x_{opt})$, $g(x_{opt})$ and $h(x_{opt})$ precise values are computed at the calculated points at the x_{opt} set points solutions set obtained in this manner. If the termination condition for calculations is not met, all values obtained from the exact models are added to the training sample: $x_s = x_s + x_{opt}$, $f(x_s) = f(x_s) + f(x_{opt})$, $g(x_s) = g(x_s) + g(x_{opt})$, $h(x_s) = h(x_s) + h(x_{opt})$.
5	Proceeding back to stage 2 involves rebuilding approximate models once more.
6	The criteria for terminating calculations are established: • the constructed models overall relative error e achieves a specified minimum: $e = \frac{1}{k \cdot (n+p+q)} \cdot \sum_{j=1}^{n+p+q} \sqrt{\sum_{i=1}^{k} \left(\frac{M_{ij}(x) - F_{ij}(x)}{F_{ij}(x)} \right)^2} \le \varepsilon,$ where k represents the solutions number in the identified Pareto-optimal set; $M_{ij}(x)$ denotes the functions $f(x)$, $g(x)$ or $h(x)$ one value, determined using its approximate model; $F_{ij}(x)$ represents the same function derived value from the exact model, and ε is a sufficiently small positive number. Meeting this criterion indicates that the constructed approximate models quality allows them to be substituted for the exact models effectively; • identifying one or more vectors $f(x)$ that meet the specified criteria $f(x) \le f_{expert}$ given the constraints $g(x) > 0$ and $h(x) = 0$, where f_{expert} denotes objective function values provided by experts, sufficient to guarantee the required product characteristics; • reaching the computations maximum allowable number with accurate models; • exceeding the permissible computation time limit.

At the first stage, the training sample (table 3) is formed according to the helicopter TE parameters (on the example, the TV3-117 TE [49]), which are registered on board the helicopter: the gas generator rotor r.p.m. (n_{TC}), the free turbine rotor speed (n_{FT}), the gas temperature in front of the compressor turbine (T_G), as well as atmospheric parameters (h is the flight altitude, T_N is the air temperature, P_N is the air pressure, ρ is the air density), reduced to absolute values according to gas-dynamic similarity theory [50, 51]. The input data detailed analysis and preliminary processing is given in [52–55].

During the input data initial preprocessing stage, the Fisher-Pearson test [56] is computed using the observed frequencies m_1, …, m_r (which sum up each measurement result probabilities across each category), and it is compared against the Fisher-Pearson test χ^2 critical values with freedom degrees $r - k - 1$. With $r - k - 1 = 13$ and $\alpha = 0.05$, the computed random variable $\chi^2 = 3.588$ did not exceed the critical value is 22.362. This indicates that the hypothesis of normal distribution can be accepted, confirming sample homogeneity. Further validation comes from the Fisher-Snedecor test [57], where the larger variance ratio to the smaller one is 1.28 (below the standardized critical value $F_{critical} = 3.44$). Thus, it can be concluded that the training sample is homogeneous.

To ensure the training and test datasets representativeness, initial data underwent cluster analysis (table 3), resulting in the eight distinct categories identification (fig. 5, a). Following a randomization procedure, training (control) and test samples were allocated in a 2:1 ratio (67 % and 33 %, respectively). The training both clustering analysis (fig. 5, b) and test samples confirmed their composition into the same eight categories observed in the control sample. The distances similarity between categories across all samples underscores the both training and test datasets representativeness [52–55]. Detailed findings from the research are summarized in table 4.

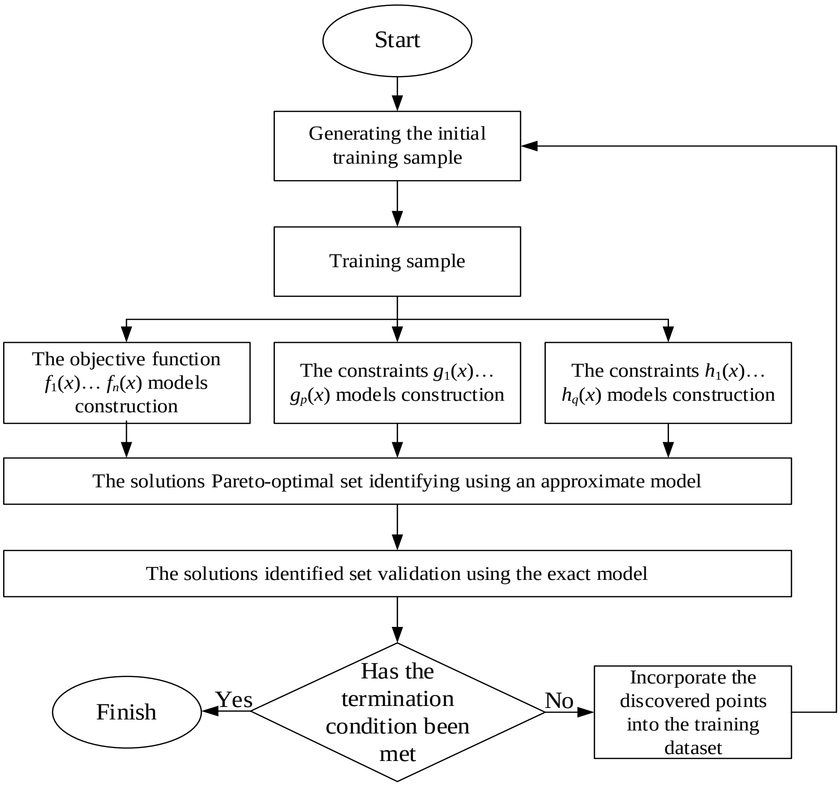
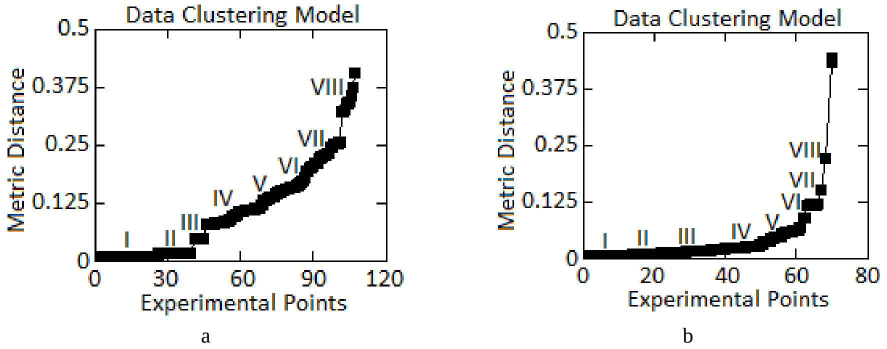

Start
Generating the initial training sample
Training sample
The objective function $f_1(x)$… $f_n(x)$ models construction
The constraints $g_1(x)$… $g_p(x)$ models construction
The constraints $h_1(x)$… $h_q(x)$ models construction
The solutions Pareto-optimal set identifying using an approximate model
The solutions identified set validation using the exact model
Has the termination condition been met
Yes
Finish
No
Incorporate the discovered points into the training dataset

Fig.4. Algorithm of the multi-criteria optimization method using heuristic approaches – evolutionary and genetic algorithms [34, 46]

Table 3. Training sample fragment (author's research, published in [52–55])

Number	Gas generator rotor r.p.m., n_{TC}	Free turbine rotor speed, n_{FT}	Gas temperature in front of the compressor turbine, T_G
1	0.929	0.943	0.932
2	0.933	0.982	0.964
3	0.952	0.962	0.917
4	0.988	0.987	0.908
…	…	…	…
256	0.973	0.981	0.953



a

b

Fig.5. Diagrams of clustering results (highlighted I...VIII classes) on initial experimental (a) and training (b) samples (author's research, published in [52–55])

Table 4. The results of determining the optimal amounts of training and test samples (author's research)

Sample	Application	Amounts
Training	It is employed for model training to address practical issues using the available data.	256 elements (100 %)
Test	It is utilized to verify the constructed model and the control sample adequacy.	84 elements (the training sample 33 %)
Control	It is employed to oversee the neural network's training process and is deemed adequate to ensure its representativeness.	172 elements (the training sample67 %)

In the computational experiment subsequent stage, based on the input sample data, the dependency describing the increase in total pressure in the compressor is approximated [46]:

$$f\left(x_1, x_2\right) = \frac{x_1^{\frac{k-1}{k}} - 1}{x_1^{\frac{k-1}{k} \cdot \frac{1}{x_2}} - 1}, \tag{18}$$

where $k = 1.4$ is the adiabatic exponent physical meaning, and variables change within $0 \leq x_1 \leq 20$ and $0 \leq x_2 \leq 20$.

The selected expression serves as a benchmark due to its role in the helicopter TE compressor efficiency calculating, a critical factor in engine performance assessment. An experiment generated 625 data pairs ($[x_1, x_2]$, d) were generated with variables x_1 and x_2 varied uniformly. A neural network 2–36–1 configuration was utilized, featuring 2 input neurons, 36 Gaussian radial neurons, and one output neuron. A dedicated training algorithm was employed, achieving a maximum approximation error is 0.06 after 200 iterations. The same training data was used across 20 iterations to neural network training with 26 radial neurons, reducing the approximation error to 0.02 for the polymorphic RBF network. Fig. 6(a) depicts the function's representation diagram, while fig. 6(b) illustrates the associated approximation error. Hence, the proposed method for constructing polymorphic RBF networks has the potential to significantly reduce computational time and produce more efficient networks (with fewer neurons and lower error) compared to traditional approaches.

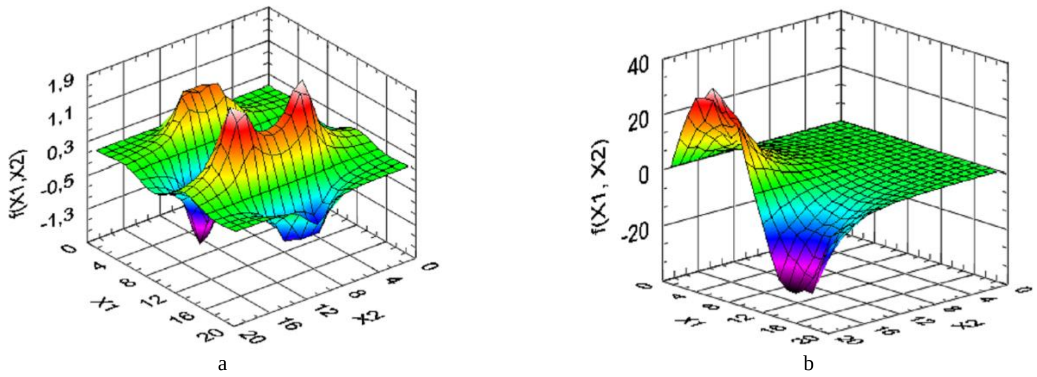


Fig.6. The 3D surfaces depicting the approximated function are as follows: (a) – illustration of the test function, (b) – representation of the error in approximating the test function (author's research, published in [46])

Alternative models for function (18) were additionally constructed using well-established and commonly employed methodologies, including the multilayer perceptron, cascade correlation network, and the group argument consideration method. The complete dataset is 625 entries was randomly partitioned into training (comprising the entries 90 %) and testing (consisting the entries 10 %) subsets. This ensured constructed models' comprehensive evaluation and validation.

Table 5. Experimental sampling and training results of a polymorphic RBF network (author's research)

Number	Gas temperature in front of the compressor turbine, T_G (x_{train})	Degree of increase in total pressure in the compressor (y_{train}) obtained by (18)	Degree of increase in total pressure in the compressor ($y_{neural\ network}$) obtained by polymorphic RBF network	$\Delta y = \mid y_{train} - y_{neural\ network} \mid$
1	0.932	0.860394	0.857214	0.00318 (0.32 %)
2	0.964	0.830571	0.833681	0.00311 (0.31 %)
3	0.917	0.895151	0.892011	0.00314 (0.31 %)
4	0.908	0.841192	0.844352	0.00316 (0.32 %)
…	…	…	…	…
32	0.924	0.881076	0.877886	0.00319 (0.32 %)

In the computational experiment subsequent phase, the polymorphic RBF network training was conducted using an experimental dataset value consisting representing the total pressure increase degree in the compressor and the gas temperature in front of the compressor turbine. The training dataset comprised 256 values, corresponding to the experimental parameters mentioned. From this dataset, 36 nodes were selected as centers for the radial basis functions. The network output, representing a one-dimensional relation, was trained using 32 specific examples detailed in table 5. A saturation parameter value is 0.3 was employed to configure the polymorphic RBF network [28].

As can be seen from the table 6, the full pressure degree error obtained values increases in the compressor Δy, obtained according to the analytical expression (18) y_{train} and using the polymorphic RBF network $y_{neural\ network}$, does not exceed 0.32 %, which indicates the proposed polymorphic RBF network high ability to approximate complex analytical dependencies. When reducing the hidden layer neurons number from 36 to 20, error Δy almost did not increase (increased by 0.08 %). The comparative analysis results of the polymorphic RBF network training with other RBF network architectures with additive interference in the white noise with zero mathematical expectation σ_i = 0.025 form, which corresponds to 2.5 % (classical, RBF network with a separate hidden layer for each independent variable and a radial basis functions in each group different number [27], RBF perceptron [28]) of the same structures 2–36–1 are given in table 6.

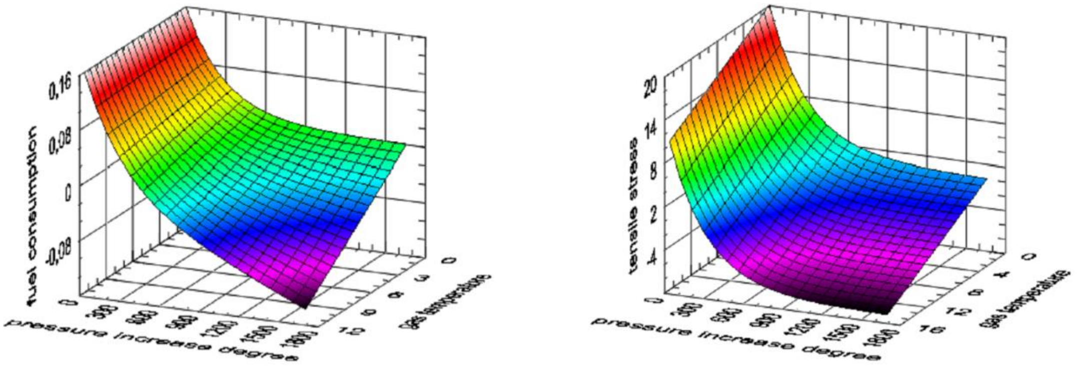
Table 6. The results of a comparative analysis of RBF networks training of different architectures (author's research)

Characteristic	Classical RBF network	RBF network with a separate hidden layer for each independent variable and a different number of radial basis functions in each group	RBF perceptron	Polymorphic RBF network
Training error, %	6.64	4.87	0.45	0.38
Testing error, %	7.38	6.11	7.43	0.73
Training time, ms	1.0	1.0	1.0	1.0

As can be seen from the table 6, the polymorphic RBF network training error is 0.38 %, which is 17.5 times lower than the classical RBF network training error, 12.8 times lower than the RBF network with a separate hidden layer for each independent variable and a radial basis functions different number in each group [27] training error, is 1.2 times lower than the RBF-perceptron network [28] training error. Also from the table 6, it can be seen that the polymorphic RBF network testing error is 0.73 %, which is 10.1 times lower than the classical RBF network testing error, 8.4 times lower than the RBF network with a separate hidden layer for each independent variable and the radial basis functions different number in each group [27] testing error, is 10.2 times lower than the RBF-perceptron network [28] testing error. The specified architectures RBF networks training time is the same and is 1 ms. Thus, the key advantage in choosing a polymorphic RBF network for the helicopter TE working process parameters optimizing task solving is the training and testing error lowest values.

4. Results

In order to check the polymorphic RBF network effectiveness in the helicopter TE working process parameters optimizing task (on the example, the TV3-117 TE), seven relevant independent parameters are defined in [46]: π_C^*, T_G^*, λ_A, λ_C, λ_G, λ_T, λ_{FT} are the gas by air the inlet section, compressor, combustion chamber, compressor turbine and free turbine, respectively. When choosing an independent parameters admissible combination, the restrictions are h_z is the last compressor stage blade height and σ_p is the tensile stress in turbine last stage impeller blade. This model allows you to vary the π_C^* and T_G^* values to obtain the helicopter TE working process optimal parameters. The $C_{spec.\ value}$ (kg/N·h), and σ_p (kg/mm^2) on π_C^* and T_G^* (K) dependences are shown in fig. 7, a and b, respectively.



a

b

Fig.7. Diagram illustrating the independent variables impact on objective functions (a) and constraints (b) (author's research)

In seeking the TV3-117 TE Pareto-optimal operating regime, the objective is to minimize variables such as specific fuel consumption $\pi_C^* = 4...20$, $T_G^* = 1300...1800$ K, given gas flow rates $\lambda_A = 0.6...0.7$, $\lambda_K = 0.25...0.35$, $\lambda_G = 0.15...0.25$, $\lambda_T = 0.4...0.65$, $\lambda_{FT} = 0.5...0.7$ and restrictions: the compressor last stage blade height $h_z > 15$ mm, tensile stress in the turbine last stage blade $\sigma_p > 25$ kg/mm^2 and $\varepsilon = 0.005$ under the calculations completion conditions (table 2, stage 6). The Pareto-optimal set finding task solving process according to the polymorphic RBF network developed training algorithm is given in the table 7. The initial training sample comprises 45 solution vectors $\mathbf{x} = (\pi_C^*, T_G^*, \lambda_A, \lambda_C, \lambda_G, \lambda_T, \lambda_{FT})$ generated using a central composite design with centers positioned on the faces [46]. Among these 45 solutions, 9 meet the constraint criteria, and 7 are non-dominated. Table 7 presents the neurons number N_h in the hidden layer and the fitness e_m for the target variables each approximate model and constraints using the polymorphic RBF network approach described earlier.

Table 7. The results of a comparative analysis of RBF networks training of different architectures (author's research, published in [46])

Iterations		First iteration	Second iteration	Third iteration	Results
The equations number in the training sample		50	140	230	320
The equations number satisfying the constraint		10	45	110	200
Pareto optimal set size		8	15	24	40
Model $C_{spec.\ value}$	N_h	37	39	35	–
	e_m	0.00009	0.00011	0.00007	–
Model σ_p	N_h	37	40	38	–
	e_m	0.02542	0.01768	0.34325	–
Model h_z	N_h	40	35	36	–
	e_m	0.00005	0.00004	0.00004	–
Models total relative error	e	0.0065	0.0054	0.0037	–

Using the NSGA-II algorithm [46, 58], 100 Pareto-optimal solutions were identified with a total relative error $e = 0.0065$ over 500 training generations with a population size of 100 individuals. After validation against the exact model, these solutions were integrated into the training sample, resulting in a 140 vectors total (including 45 satisfying constraints and 15 from the Pareto-optimal set). This marked the first iteration completion. Subsequent iterations followed, with a total of three iterations conducted and 320 function minimization calls made. The relative errors for the models in the second and third iterations were $e = 0.0054$ and $e = 0.0037$ respectively. Table 8 displays the TV3-117 TE found select Pareto-optimal operational parameters.

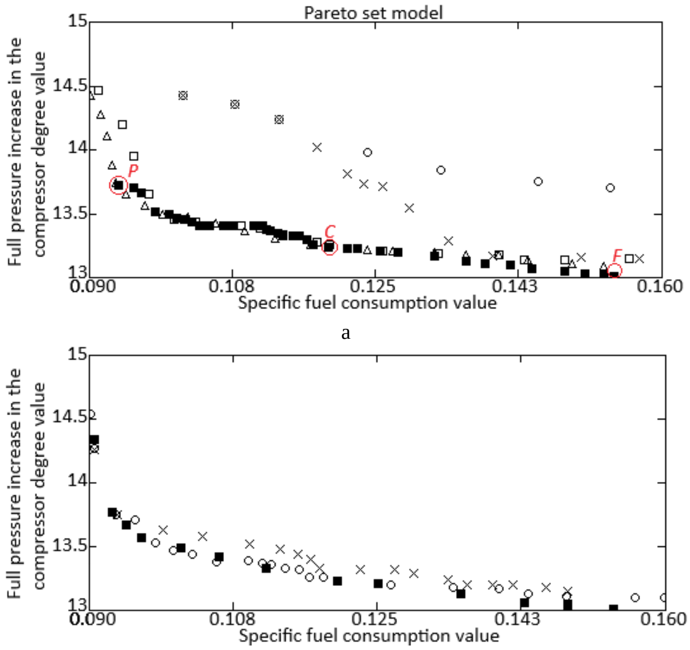
Table 8. An example of the TV3-117 turboshaft engine working process parameters variant at the helicopter flight mode (author's research, published in [45])

$C_{spec.\ value}$, kg/N·h	σ_p, kg/mm^2	h_z, mm	π_C^*	T_G^*, K	λ_A	λ_C	λ_G	λ_T	λ_{FT}
0.085	12.9	15	13.0	1310	0.685	0.250	0.25	0.640	0.682
0.089	17.4	15	10.9	1302	0.693	0.278	0.25	0.462	0.577
0.092	17.7	15	10.2	1308	0.693	0.299	0.25	0.468	0.581
0.096	17.5	15	8.7	1308	0.693	0.343	0.25	0.507	0.586
0.101	16.9	15	8.6	1354	0.693	0.343	0.25	0.539	0.593
0.109	17.0	15	8.7	1466	0.693	0.343	0.25	0.525	0.599
0.115	17.1	15	8.6	1545	0.693	0.343	0.25	0.508	0.575
0.124	17.7	15	8.2	1638	0.693	0.343	0.25	0.485	0.564
0.130	17.2	16	8.0	1699	0.693	0.343	0.25	0.505	0.542
0.137	21.5	19	6.9	1684	0.693	0.343	0.25	0.443	0.529
0.146	24.2	23	5.3	1686	0.693	0.343	0.25	0.469	0.511

The results of all iterations are shown in fig. 8, a ("○" is the initial training sample (50/8), "×" is the first iteration (140/15), "□" is the second iteration (230/24), "Δ" is the third iteration (320/40); "■" is the decision based on the exact model (50000/100)). This figure also shows the Pareto-optimal set (Pareto front) obtained by the NSGA-II method [46, 58] (with 100 individuals and 500 generations population size) based on the exact model. To find this set, 50000 calls of minimization functions were made.

In fig. 8, b ("○" is the approximate model, "×" is the exact model: 5 generations, "■" is the exact model: 500 generations) compares three Pareto-optimal solutions sets: obtained on the proposed approximate model basis (with 320 calls of the exact model), as well as sets obtained on the exact model basis with 500 calls (with 100 individuals and 5 generations population size) and 50000 calls (with 100 individuals and 500 generations population size).

The compressor total pressure ratio and specific fuel consumption values are normalized. Target functions can be optimized for the fuel injection parameters certain values and engine control law choice. Since there is always a trade-off between optimization aims, the optimal injection parameter values final choice must be determined by the operating engine conditions. So, if the aim is to minimize specific fuel consumption, then you should select the engine operating mode, which is determined by point *P*, but in this case the total pressure increase in the compressor value will increase by 5.0 %. However, in this case there is a damage risk to the compressor blades and, as a result, a decrease in engine performance and the emergency situations risk. If the aim is to minimize specific effective fuel consumption, then the engine operating mode will be determined by point *F*, with the total pressure increase in the compressor reduced value by 1.0 %. A compromise solution will be the engine operating mode, which is determined by point *C*, where the degree of increase in total pressure in the compressor is normal, while the specific fuel consumption increases insignificantly compared to the value at point *F* – by 10.5 %, which is unimportant for helicopters TE. Since the helicopter crew commander knows in advance which of the criteria interests him more, individual solutions that are optimal according to the most significant criteria are considered on the resulting Pareto front [59, 60]. Thus, the helicopter crew commander receives support in decision making on the engine operating mode choice: with the specific fuel consumption minimum value, with a maximum degree of increase in total pressure in the compressor, or a compromise option depending on the helicopter flight conditions.



b

Fig.8. Pareto-optimal sets evolution of solutions in the calculations process (a) and three Pareto-optimal sets of solutions comparisons (b) (author's research)

5. Discussion

To evaluate the multicriteria optimization task solving results effectiveness based on the created polymorphic RBF network, comparative computational experiments were carried out using a self-organizing multiobjective evolutionary algorithm (SMEA) [61] and a multiobjective differential evolutionary self-learning algorithm (Multioobjective Differential Evolutionary Self-Learning algorithm, MDESL) [61, 62]. The SMEA algorithm uses self-organizing maps (SOM) during the training process to establish neighborhoods. The MDESL algorithm for extracting the neighborhood relations knowledge uses *k*-means clustering for subsequent feature pairing. The *k*-means clustering algorithm comes with high computational costs due to the iterative strategy use. Test sets TEST_1 – TEST_8, each with the TV3-117 TE 32 values parameters, were used as reference tasks. Test sets TEST_1 – TEST_8 was obtained by dividing the total training sample (table 2) into 8 equal parts. To conduct comparative computational experiments, the following parameter settings were chosen for each of the algorithms, taking into account the test datasets TEST_1 – TEST_8 features:

1. General parameters: population size N = 500 for two-criteria optimization, variables (objects) dimension n = 8, evolutionary generations maximum number T = 320, test runs number is 32 each test instance runs, stopping condition – all algorithms stop execution after 50000 function calculations.
2. The multicriteria optimization algorithm using heuristic approaches parameters is the evolutionary and genetic algorithms (fig. 4) using the created polymorphic RBF network: components number K_{max} = 8, use probability e = 0.6, crossover constant CR = 1 and differentiation constant F = 0.7 [61, 63].
3. SMEA algorithm parameters: SOM structure is the 1-D map dimension, neurons number 1*100 for two-criteria optimization, initial training rate: τ_0 = 0.9, neighborhood size T = 5, pairing restriction probability β = 0.7.
4. Parameters of the MDESL algorithm [61, 63]: clusters maximum number: K_{max} = 10, history length H = 5,

crossover constant *CR* = 1 and differentiation constant *F* = 0.6.

To quantify the Pareto fronts quality obtained as the approximations result, two quality indicators were used: inverted intergenerational distance (*IGD*) and hypervolume (*HV*) [61, 64]. The optimality guideline is represented by the vector $r^* = \left(r_1^*, r_2^*, ..., r_k^*\right)$, where $r_j^* = \max f_j\left(x\right) + 1$, *j* = 1...*k*, *k* is the task objective functions number. According to [57], decreasing the *IGD* metric value improves the *P* approximation in the real Pareto front P^* approximation, and the *IGD* metric is determined according to the expression:

$$IGD\left(P^*, P\right) = \frac{\sum\limits_{x^* \in P^*} d\left(x^*, P\right)}{\left|P^*\right|}, \tag{19}$$

where *P* is the approximate approximated Pareto front, P^* is the optimal points set in the true Pareto front, $d(x^*, P)$ is the minimum distance between uniformly distributed optimal points x^* and any approximation point *P*, $|P^*|$ is the power P^*.

The work also used the *HV* metric, increasing which value improves the *P* approximation in the real Pareto front P^* approximation. According to [61], the *HV* metric is calculated as the difference between the approximated points *P* and the reference points r^* in the target space according to the expression:

$$HV = \left(P, r^*\right) = VOL\left(\bigcup_{x \in P}\left(f_1\left(x\right), r_1\right) \times ... \times \left(f_k\left(x\right), r_k\right)\right), \tag{20}$$

where $r^* = \left(r_1^*, r_2^*, ..., r_k^*\right)$ is the optimality reference point in the target space, dominant for any Pareto-optimal point, *VOL*(•) is the Lebesgue measure.

The proposed multicriteria optimization algorithm using heuristic approaches are the evolutionary and genetic algorithms (fig. 4) using the created polymorphic RBF network was compared with the SMEA and MDESL algorithms, each of which was run 32 times independently on given test sets, and the *IGD* and *HV* metrics were calculated, which average values are given in tables 9 and 10 respectively.

Table 9. The *IGD* metric average values calculating results after 32 independent runs of test sets TEST_1 – TEST_8 (author's research, based on [57])

Number	Test dataset	*IGD* metric average value		
		Proposed algorithm	SMEA	MDESL
1	TEST_1	1.168	0.876	2.987
2	TEST_2	1.225	0.924	3.118
3	TEST_3	1.217	0.899	3.064
4	TEST_4	1.435	1.022	3.522
5	TEST_5	1.601	1.113	3.759
6	TEST_6	1.484	1.095	3.602
7	TEST_7	1.795	1.192	4.017
8	TEST_8	1.652	1.156	3.838

Table 10. Results of calculating the average values of the *HV* metric after 32 independent runs of test sets TEST_1 – TEST_8 (author's research, based on [57])

Number	Test dataset	*HV* metric average value		
		Proposed algorithm	SMEA	MDESL
1	TEST_1	5.622	3.832	1.922
2	TEST_2	5.537	3.693	1.805
3	TEST_3	5.558	3.713	1.818
4	TEST_4	5.386	3.407	1.654
5	TEST_5	5.117	3.235	1.588
6	TEST_6	5.266	3.303	1.621
7	TEST_7	5.026	3.092	1.377
8	TEST_8	5.074	3.101	1.462

Tables 9 and 10 show that the proposed multicriteria optimization algorithm using heuristic approaches are the evolutionary and genetic algorithms (fig. 4) with the polymorphic RBF network are the outperformed the classical SMEA and MDESL algorithms across all eight test sets (TEST_1 – TEST_8) according to the IGD and HV metrics. The proposed algorithm consistently showed superior performance, while the SMEA and MDESL algorithms achieved average or low results. Additionally, the approximation edges produced by the proposed algorithm displayed the best diversity and closeness to real Pareto fronts among all three algorithms tested [65–67].

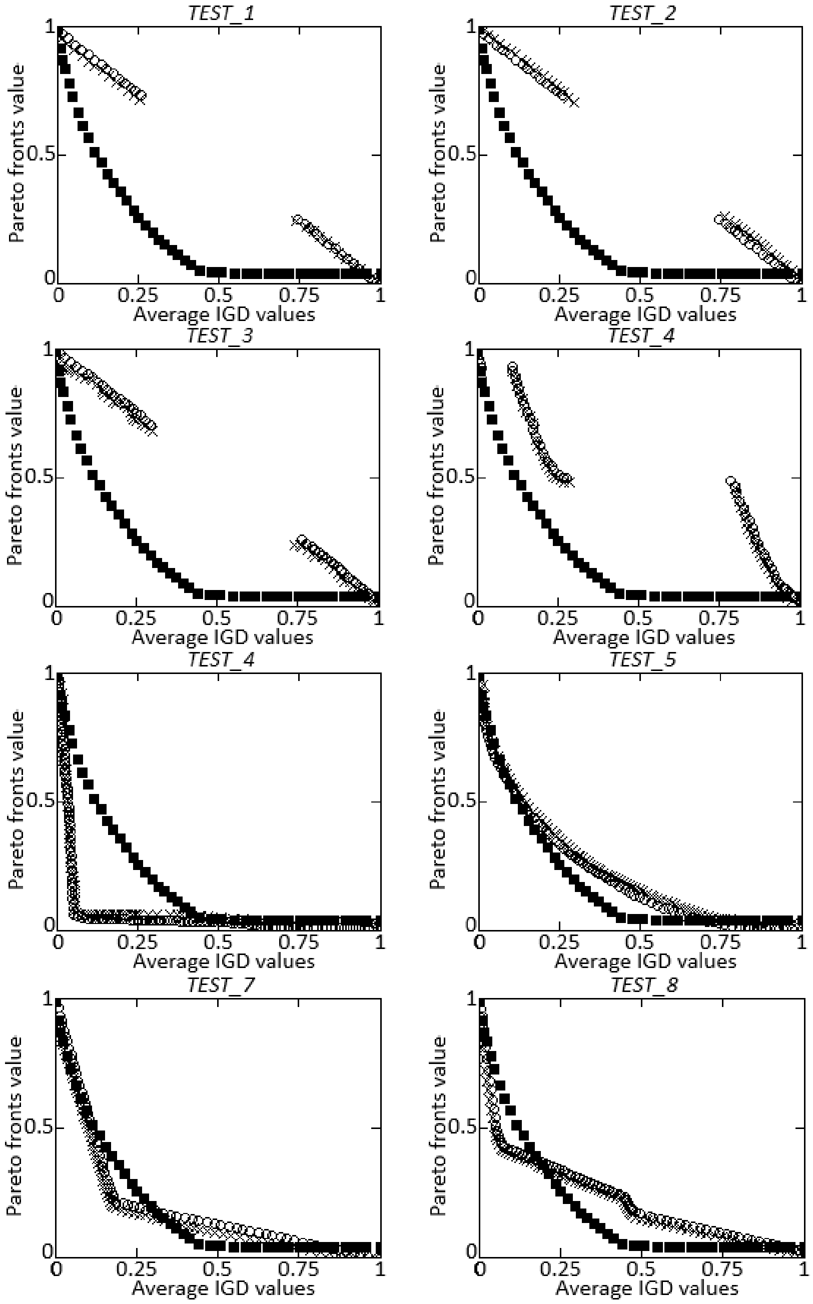


Fig.9. Resulting diagrams of the calculated Pareto-front approximations with average values of the *IGD* metric in 32 independent runs of test sets TEST_1 – TEST_8 for the proposed algorithm and the SMEA algorithm (author's research based on [54])

Fig. 9 illustrates the approximating Pareto fronts with average IGD values for the proposed algorithm and the SMEA algorithm. The proposed algorithm's fronts converge closely to the real Pareto fronts in all test cases, while the SMEA algorithm's fronts fail to converge completely on datasets TEST_5 – TEST_8 and have missing segments on TEST_1 – TEST_4 datasets.

Fig. 10 shows the IGD efficiency indicator against the evolutionary generations number. The proposed algorithm required fewer calculations than the other algorithms, demonstrating superior search efficiency across all test sets TEST_4 – TEST_8. This confirms the proposed algorithm's advantage over the SMEA and MDESL algorithms.

Thus, the classical evolutionary algorithms SMEA and MDESL with the proposed multicriteria optimization algorithm comparison using heuristic approaches are the evolutionary and genetic algorithms (fig. 4) using the created polymorphic RBF network demonstrated the latter significant superiority in convergence and diversity terms, measured by metric values *IGD* and *HV*, at lower computational costs.

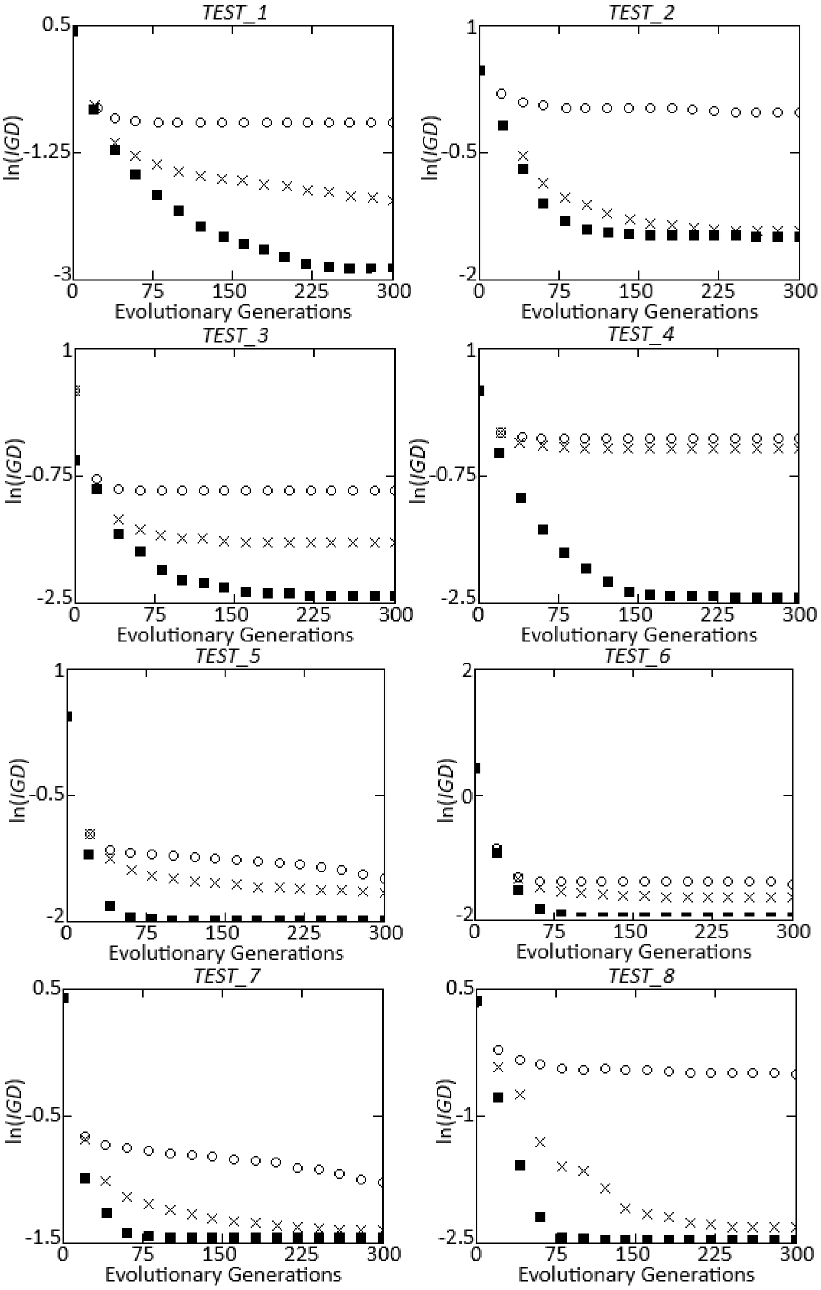


Fig.10. Resulting diagrams showing the evolution of the average *IGD* metric values and corresponding standard deviations in the test suites TEST_1 – TEST_8 (author's research based on [54])

6. Conclusions

1. For the first time, the RBF network new architecture was created is a polymorphic RBF network, which differs from the traditional RBF network in that due to a separate hidden layer for each independent variable and a different number of multidimensional radial basis functions in each group and the radial elements summation multiplication element instead use allows to eliminate its shortcomings and limitations, namely:

- better generalization ability: the polymorphic RBF network generalizes better on new data, thanks to which it is more resistant to retraining. It has been experimentally confirmed that the newly created polymorphic RBF network training occurs immediately, deterministically, in the computational procedure one cycle, while its training time is 1 ms. At the same time, its training and testing errors are 0.38 and 0.73, respectively, which are the lowest among other RBF network architectures under the same training conditions;
- increased robustness to noise: the polymorphic RBF network is more robust to noise in the training data. It is experimentally confirmed that the polymorphic RBF network training and testing errors are 17.5 and 10.1 times lower, respectively, than the classical RBF network training and testing errors, and are 12.8 and 8.4 times lower, respectively, than the RBF network with a separate hidden layer for each independent variable and a radial basis

functions in each group different number [27] training and testing errors, and are 1.2 and 10.2 times lower, respectively, than the RBF-perceptron network [28] training and testing errors;

- computational complexity reduction: the polymorphic RBF network features a dynamically structured hidden layer, resulting in lower training computational complexity. Experimentally, it was confirmed that the approximation error for complex analytical dependencies (e.g., the relations between the TV3-117 turboshaft engine total pressure increase in the compressor and gas temperature in front of the compressor turbine) by a polymorphic RBF network with a 2–36–1 structure does not exceed 0.32 % compared to analytical calculations. The hidden layer neurons number reducing to 20 only slightly increased the error by 0.08 %.

2. For the first time, a polymorphic RBF network training algorithm was created, which basis is an evolutionary algorithm for training classic RBF networks, which due to the weight coefficients initialization methods use taking into account the tasks and previous values structure, the tournament selection for mutations use, the additional criteria addition to the fitness function in order to take into account the polymorphic RBF network training stability and speed, as well as the mutation evolutionary strategy application, allowed to obtain the polymorphic RBF network lowest training and testing errors compared to the classical RBF network, the RBF network with a separate hidden layer for each independent variable and a radial basis functions in each group different number [27], by the RBF-perceptron network [28]. It was experimentally confirmed that the 2–36–1 structure polymorphic RBF network training and testing errors were 0.38 and 0.73 %, respectively, while the classical RBF network training and testing errors were 6.64 for the similar 2–36–1 structure and 7.38 %, RBF networks with a separate hidden layer for each independent variable and a radial basis functions in each group different number [27] are the 4.87 and 6.11 %, RBF-perceptron networks [28] are the 0.45 and 7.43 %.

3. The newly created polymorphic RBF network practical application possibility and its training algorithm is experimentally shown on the example the helicopter TE (on the example of the TV3-117 turboshaft engine) working process parameters optimizing task solving according to the multi-criteria optimization algorithm:

- the tensile stress spatial dependences in the turbine last stage impeller blade and engine specific fuel consumption were obtained, depending on the gas temperature in front of the compressor turbine and the degree of increase in the total pressure in the compressor values;
- the optimal Pareto front was obtained, which made it possible to obtain the engine operation three additional modes: maximum reduction of specific fuel consumption at the total pressure in the compressor increase degree increased value by 5.0 %, specific fuel consumption minimization at the total pressure in the compressor increase degree reduced value by 1.0 %, the total pressure in the compressor increases degree optimal value with a slight increase in specific fuel consumption by 10.5 %;
- the TV3-117 turboshaft engine working process parameters experimental version at the helicopter flight mode was obtained, which consists the specific fuel consumption, the total pressure in the compressor increase degree and the specified gas flow rates optimal values at the inlet device, compressor, combustion chamber, compressor turbine and free turbine, respectively, the use of which will allow the helicopter crew commander to adjust the engine operating modes during the helicopter flight and thereby increase flight safety;
- it was experimentally confirmed that the newly created polymorphic RBF network use and its training algorithm in the TV3-117 turboshaft engine working process parameters optimizing task at the helicopter flight mode according to the multi-criteria optimization algorithm compared to the self-organizing multi-objective evolutionary algorithm (SMEA) and the multi-criteria differential evolutionary self-learning algorithm (MDESL) [61, 62], is the most effective, as evidenced by the IGD metric minimum values and the HV metric maximum values. When using the polymorphic RBF network and its training algorithm, the IGD metric values are within 1.168...1.795, while when using the SMEA algorithm, its values are within 0.876...1.192, when using the MDESL algorithm are within 2.987...4.017. When using the polymorphic RBF network and its training algorithm, the HV metric values are within 5.026...5.622, while when using the SMEA algorithm, its values are within 3.092...3.832, when using the MDESL algorithm are within 1.377...1.922.

Prospects for further research are the developed methods and models adaptation into the helicopter turboshaft engines operation monitoring and controlling at flight operation mode general concept. This concept is implemented in the neural network expert system [68,69] and the helicopter turboshaft engines closed neural network on-board automatic control system [48].

References

[1] J. Bi, Z. Wang, H. Yuan, J. Zhang, and M. C. Zhou, “Self-adaptive teaching-learning-based optimizer with improved RBF and sparse autoencoder for high-dimensional problems”, *Information Sciences*, vol. 630, pp. 463–481, 2023. https://doi.org/10.1016/j.ins.2023.02.044

[2] W. Jia, D. Zhao, and L. Ding, “An optimized RBF neural network algorithm based on partial least squares and genetic algorithm for classification of small sample”, *Applied Soft Computing*, vol. 48, pp. 373–384, 2016. https://doi.org/10.1016/j.asoc.2016.07.037

[3] Z. Liu, C.-S. Leung, and H. Cheung So, “Formal convergence analysis on deterministic l_1-regularization based mini-batch

learning for RBF networks", *Neurocomputing*, vol. 532, pp. 77–93, 2023. https://doi.org/10.1016/j.neucom.2023.02.012

[4] Z. Han, X. Qian, H. Huang, and T. Huang, "Efficient design of multicolumn RBF networks", *Neurocomputing*, vol. 450, pp. 253–253, 2021. https://doi.org/10.1016/j.neucom.2021.04.040

[5] R. Vaish, S. Tewari, S.M. Tripathi, and U.D. Dwivedi, "Machine Learning Applications in Power System Fault Diagnosis: Research Advancements and Perspectives", *Engineering Applications of Artificial Intelligence*, vol. 106, 104504, 2021. https://doi.org/10.1016/j.engappai.2021.104504

[6] A. Panaye, B.T. Fan, J.P. Doucet, X.J. Yao, R.S. Zhang, M.C. Liu, Z.D. Hu, "Quantitative structure-toxicity relationships (QSTRs): a comparative study of various non linear methods. General regression neural network, radial basis function neural network and support vector machine in predicting toxicity of nitro- and cyano- aromatics to Tetrahymena pyriformis", *SAR and QSAR in Environmental Research*, vol. 17, 75–91, 2006. https://doi.org/10.1080/10659360600562079

[7] N.K.F. Khah, B. Salehi, P. Kianoush, and S. Varkouhi, "Estimating elastic properties of sediments by pseudo-wells generation utilizing simulated annealing optimization method", *Results in Earth Sciences*, vol. 2, 100024, 2024. https://doi.org/10.1016/j.rines.2024.100024

[8] Y. Wu, H. Wang, B. Zhang, and K.-L. Du, "Using Radial Basis Function Networks for Function Approximation and Classification", *International Scholarly Research Notices*, vol. 2012, 324194, 2012. https://doi.org/10.5402/2012/324194

[9] K.-L. Du, and M.N.s. Swamy, "Radial Basis Function Networks", *Neural Networks and Statistical Learning*, pp. 299–335, 2014. https://doi.org/10.1007/978-1-4471-5571-3_10

[10] A. Ismayilova, and M. Ismayilov, "On the universal approximation property of radial basis function neural networks", *Annals of Mathematics and Artificial Intelligence*, 2023. https://doi.org/10.1007/s10472-023-09901-x

[11] D. S. Soper, "Using an Opportunity Matrix to Select Centers for RBF Neural Networks", *Algorithms*, vol. 16, iss. 10, 455, 2023. https://doi.org/10.3390/a16100455

[12] M. Gan, H. Peng, and X.-p. Dong, "A hybrid algorithm to optimize RBF network architecture and parameters for nonlinear time series prediction", *Applied Mathematical Modelling*, vol. 36, pp. 2911–2919, 2012. https://doi.org/10.1016/j.apm.2011.09.066

[13] D. Xu, Z. Hui, Y. Liu, and G. Chen, "Morphing control of a new bionic morphing UAV with deep reinforcement learning", *Aerospace Science and Technology*, vol. 92, pp. 232–243, 2019. https://doi.org/10.1016/j.ast.2019.05.058

[14] Q. Junfei, M. Xi, and L. Wenjing, "An incremental neuronal-activity-based RBF neural network for nonlinear system modeling", *Neurocomputing*, vol. 302, pp. 1–11, 2018. https://doi.org/10.1016/j.neucom.2018.01.001

[15] S. Ding, L. Xu, C. Su, and F. Jin, "An optimizing method of RBF neural network based on genetic algorithm", *Neural Computing and Applications*, vol. 21, iss. 2, pp. 333–336, 2012. https://doi.org/10.1007/s00521-011-0702-7

[16] F. J. Pontes, A. P. de Paiva, P. P. Balestrassi, J. R. Ferreira, and M. B. da Silva, "Optimization of Radial Basis Function neural network employed for predicition of surface roughness in hard turning process using Taguchi's orthogonal arrays", *Expert Systems with Applications*, vol. 39, iss. 9, pp. 7776–7787, 2012. https://doi.org/10.1016/j.eswa.2012.01.058

[17] Q.-Y. Chen, L. Chen, J.-N. Su, M.-J. Fu, and G.-Y. Chen, "Model selection for RBF-ARX models", *Applied Soft Computing*, vol. 121, 108723, 2022. https://doi.org/10.1016/j.asoc.2022.108723

[18] D. G. B. Franco, and M. T. A. Steiner, "New Strategies for Initialization and Training of Radial Basis Function Neural Networks", *IEEE Latin America Transactions*, vol. 15, iss. 6, pp. 1182–1188, 2017. https://doi.org/10.1109/TLA.2017.7932707

[19] F. N. Mojarrad, M. H. Veiga, J. S. Hesthaven, and P. Offner, "A new variable shape parameter strategy for RBF approximation using neural networks", *Computers & Mathematics with Applications*, vol. 143, pp. 151–168, 2023. https://doi.org/10.1016/j.camwa.2023.05.005

[20] F. Ros, M. Pintore, A. Deman and J. R. Chretien, "Automatical initialization of RBF neural networks", *Chemometrics and Intelligent Laboratory Systems*, vol. 87, iss. 1, pp. 26–32, 2007. https://doi.org/10.1016/j.chemolab.2006.01.008

[21] S. Vladov, Y. Shmelov, R. Yakovliev, and M. Petchenko, "Modified Neural Network Method for Trend Analysis of Helicopter Turboshaft Engine Parameters at Flight Modes", *CEUR Workshop Proceedings*, vol. 3347, pp. 11–29, 2022.

[22] S. Kim, J. H. Im, M. Kim, J. Kim, and Y. I. Kim, "Diagnostics using a physics-based engine model in aero gas turbine engine verification tests", *Aerospace Science and Technology*, vol. 133, 108102, 2023. https://doi.org/10.1016/j.ast.2022.108102

[23] X. Li, and W. Yu, "Dynamic system identification via recurrent multilayer perceptrons", *Information Sciences*, vol. 147, iss. 1–4, pp. 45–63, 2002. https://doi.org/10.1016/S0020-0255(02)00207-4

[24] J. Struye, and S. Latre, "Hierarchical temporal memory and recurrent neural networks for time series prediction: An empirical validation and reduction to multilayer perceptrons", *Neurocomputing*, vol. 396, pp. 291–301, 2020. https://doi.org/10.1016/j.neucom.2018.09.098

[25] Q. Que, and M. Belkin, "Back to the Future: Radial Basis Function Network Revisited", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 1856–1867, 2020. https://doi.org/10.1109/TPAMI.2019.2906594

[26] D. Lowe, "Radial Basis Function Networks – Revisited", *Mathematics Today*, vol. 51, iss. 3, pp. 124–126, 2015.

[27] A. Alexandrov, and S. Dudarov, "One-dimensional alternative for multidimensional neural network of radial basis functions", *Advances in chemistry and chemical technology*, vol. 32, no. 11, pp. 13–15, 2022.

[28] I. Markin, and S. Dudarov, "An artificial neural network of radial basis functions using a perceptron as an output layer", *Advances in chemistry and chemical technology*, vol. 36, no. 11, pp. 74–76, 2022.

[29] Gh. A. Montazer, R. Sabzevari, and F. Ghorbani, "Three-phase strategy for the OSD learning method in RBF neural networks", *Neurocomputing*, vol. 72, iss. 7–9, pp. 1797–1802, 2009. https://doi.org/10.1016/j.neucom.2008.05.011

[30] Z. Geng, J. Chen, and Y. Han, "Energy Efficiency Prediction Based on PCA-FRBF Model: A Case Study of Ethylene Industries", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, issue 8, pp. 1763–1773, 2016. https://doi.org/10.1109/TSMC.2016.2523936

[31] Y. Zhao, D. Si, J. Pei, and X. Yang, "Geodesic Basis Function Neural Network", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, issue 6, pp. 8386–8400, 2022. https://doi.org/10.1109/TNNLS.2022.3227296

[32] S. Vladov, Y. Shmelov, and R. Yakovliev, "Modified Neural Network Method for Diagnostics the Helicopters Turboshaft Engines Operational Status at Flight Modes", in: Proceedings of the IEEE International Conference on System Analysis & Intelligent Computing (SAIC), Kyiv, Ukraine, October 04–07, 2022, pp. 224–229. https://doi.org/10.1109/SAIC57818.2022.9923025

[33] Y. Xi, H. Peng, and X. Chen, "A sequential learning algorithm based on adaptive particle filtering for RBF networks", *Neural*

Computing and Applications, vol. 25, issue 3–4, pp. 807–814, 2014. https://doi.org/10.1007/s00521-014-1551-y

[34] Y. Zelenkov, "Method of multi-criterial optimization based on approximate models of the researched object", *Numerical Methods and Programming*, vol. 11, pp. 250–260, 2010.

[35] V. Vichugov, "Modified gradient algorithm for training radial-basis neural networks", *Bulletin of the Tomsk Polytechnic University*, vol. 315, no. 5, pp. 149–152, 2009.

[36] R. Kumar, "Recurrent context layered radial basis function neural network for the identification of nonlinear dynamical systems", *Neurocomputing*, vol. 580, 127524, 2024. https://doi.org/10.1016/j.neucom.2024.127524

[37] N. Bezzubov, and S. Feofilov, "Application of radial-basis function networks in self-adjust adaptive systems", *System analysis, management and information processing*, no. 1, pp. 47–52, 2023.

[38] J. Maatta, V. Bazaliy, J. Kimari, F. Djurabekova, K. Nordlund, and T. Roos, "Gradient-based training and pruning of radial basis function networks with an application in materials physics", *Neural Networks*, vol. 133, pp. 123–131, 2021. https://doi.org/10.1016/j.neunet.2020.10.002

[39] L. E. Aik, W. H. Tan, and A. K. Junoh, "An Improved Radial Basis Function Networks in Networks Weights Adjustment for Training Real-World Nonlinear Datasets", *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 8(1), no. 63, pp. 63–76, 2019. http://doi.org/10.11591/ijai.v8.i1.pp63-76

[40] A. L. da Costa Oliveira, A. Britto, and R. Gusmao, "A framework for inverse surrogate modeling for fitness estimation applied to Multi-Objective Evolutionary Algorithms", *Applied Soft Computing*, vol. 146, 110672, 2023. https://doi.org/10.1016/j.asoc.2023.110672

[41] T. V. Avdeenko, K. E. Serdyukov, and Z. B. Tsydenov, "Formulation and research of new fitness function in the genetic algorithm for maximum code coverage", *Procedia Computer Science*, vol. 186, pp. 713–720, 2021. https://doi.org/10.1016/j.procs.2021.04.194

[42] L. Deng, C. Li, Y. Lan, J. Sun, and C. Shang, "Differential evolution with dynamic combination based mutation operator and two-level parameter adaptation strategy", *Expert Systems with Applications*, vol. 192, 116298, 2022. https://doi.org/10.1016/j.eswa.2021.116298

[43] S. Gupta, and R. Su, "An efficient differential evolution with fitness-based dynamic mutation strategy and control parameters", *Knowledge-Based Systems*, vol. 251, 109280, 2022. https://doi.org/10.1016/j.knosys.2022.109280

[44] H. V. H. Ayala, D. Habineza, M. Rakotondrabe, and L. Coelhod, "Nonlinear black-box system identification through coevolutionary algorithms and radial basis function artificial neural networks," *Applied Soft Computing*, vol. 87, 105990, 2020. https://doi.org/10.1016/j.asoc.2019.105990

[45] Y. Ji, Z. Yang, J. Run, and H. Li, "Multi-objective parameter optimization of turbine impeller based on RBF neural network and NSGA-II genetic algorithm", *Energy Reports*, vol. 7, suppl. 7, pp. 584–593, 2021. https://doi.org/10.1016/j.egyr.2021.10.003

[46] S. Vladov, Y. Shmelov, and R. Yakovliev, "Optimization of Helicopters Aircraft Engine Working Process Using Neural Networks Technologies", *CEUR Workshop Proceedings*, vol. 3171, pp. 1639–1656, 2022.

[47] H. Augun, "Thermodynamic, environmental and sustainability calculations of a conceptual turboshaft engine under several power settings", *Energy*, vol. 245, 123251, 2022. https://doi.org/10.1016/j.energy.2022.123251

[48] S. Vladov, Y. Shmelov, and R. Yakovliev, "Method for Forecasting of Helicopters Aircraft Engines Technical State in Flight Modes Using Neural Networks", *CEUR Workshop Proceedings*, vol. 3171, pp. 974–985, 2022.

[49] S. Vladov, Y. Shmelov, and R. Yakovliev, "Modified Method of Identification Potential Defects in Helicopters Turboshaft Engines Units Based on Prediction its Operational Status", in: Proceedings of the 2022 IEEE 4th International Conference on Modern Electrical and Energy System (MEES), Kremenchuk, Ukraine, October 20–22, 2022, pp. 556–561. https://doi.org/10.1109/MEES58014.2022.10005605

[50] S. Vladov, Y. Shmelov, and R. Yakovliev, "Modified Searchless Method for Identification of Helicopters Turboshaft Engines at Flight Modes Using Neural Networks", in: Proceedings of the IEEE 3rd KhPI Week on Advanced Technology, Kharkiv, Ukraine, October 03–07, 2022, pp. 257–262. https://doi.org/10.1109/KhPIWeek57572.2022.9916422

[51] S. Vladov, Y. Shmelov, R. Yakovliev, M. Petchenko, and S. Drozdova, "Helicopters Turboshaft Engines Parameters Identification at Flight Modes Using Neural Networks", in: Proceedings of the IEEE 17th International Conference on Computer Science and Information Technologies (CSIT), Lviv, Ukraine, November 10–12, 2022, pp. 5–8. https://doi.org/10.1109/CSIT56902.2022.10000444

[52] S. Vladov, Y. Shmelov, R. Yakovliev, and M. Petchenko, "Neural Network Method for Parametric Adaptation Helicopters Turboshaft Engines On-Board Automatic Control", *CEUR Workshop Proceedings*, vol. 3403, pp. 179–195, 2023.

[53] S. Vladov, R. Yakovliev, O. Hubachov, J. Rud, Y. Stushchanskyi, "Neural Network Modeling of Helicopters Turboshaft Engines at Flight Modes Using an Approach Based on "Black Box" Models", *CEUR Workshop Proceedings*, vol. 3624, pp. 116–135, 2024.

[54] S. Vladov, R. Yakovliev, M. Bulakh, and V. Vysotska, "Neural Network Approximation of Helicopter Turboshaft Engine Parameters for Improved Efficiency", *Energies*, vol. 17(9), 2233, 2024. https://doi.org/10.3390/en17092233

[55] S. Vladov, R. Yakovliev, V. Vysotska, M. Nazarkevych, and V. Lytvyn, "Restoring lost information method from complex dynamic object sensors based on auto-associative neural networks", *Applied System Innovation*, vol. 7(3), 53, 2024. https://doi.org/10.3390/asi7030053

[56] F. S. Corotto, "Appendix C - The method attributed to Neyman and Pearson", *Wise Use of Null Hypothesis Tests*, pp. 179–188, 2023. https://doi.org/10.1016/B978-0-323-95284-2.00012-4

[57] F. V. Motsnyi, "Analysis of Nonparametric and Parametric Criteria for Statistical Hypotheses Testing. Chapter 1. Agreement Criteria of Pearson and Kolmogorov", *Statistics of Ukraine*, no. 4'2018 (83), pp. 14–24, 2018. https://doi.org/10.31767/su.4(83)2018.04.02

[58] M. Li, H. Ma, S. Lv, L. Wang and S. Deng, "Enhanced NSGA-II-based feature selection method for high-dimensional classification", *Information Sciences*, vol. 663, 120269, 2024. https://doi.org/10.1016/j.ins.2024.120269

[59] Z. Zhang, X. Cheng, Z. Xing, and X. Gui, "Pareto multi-objective optimization of metro train energy-saving operation using improved NSGA-II algorithms", *Chaos, Solitons & Fractals*, vol. 176, 114183, 2023. https://doi.org/10.1016/j.chaos.2023.114183

[60] P. Gao, Y. Wang, H. Wang, C. Song, S. Ye, and X. Wang, "A Pareto front-based approach for constructing composite index of

sustainability without weights: A comparative study of implementations”, *Ecological Indicators*, vol. 155, 110919, 2023. https://doi.org/10.1016/j.ecolind.2023.110919

[61] N. Polkovnikova, “Hybrid expert system based on probabilistic deterministic models”, *Izvestiya SFedU. Engineering Sciences*, no. 6 (167), pp. 168–179, 2015.

[62] Q. Ye, W. Wang, G. Li, and R. Dai, “A self-organizing assisted multi-task algorithm for constrained multi-objective optimization problems”, *Information Sciences*, vol. 664, 120339, 2024. https://doi.org/10.1016/j.ins.2024.120339

[63] K.-J. Du, J.-Y. Li, H. Wang, and J. Zhang, “Multi-objective multi-criteria evolutionary algorithm for multi-objective multi-task optimization”, *Complex & Intelligent Systems*, vol. 9, pp. 1211–1228, 2023. https://doi.org/10.1007/s40747-022-00650-8

[64] T. Liguo, and S. Novikova, “Application of step-by-step training method for evolutionary algorithm in multicriterial optimization problems”, *VKGEU*, vol. 14, no. 3 (55), pp. 114–125, 2022.

[65] T. Liu, S. Song, X. Li, and L. Tan, “Approximating Pareto Optimal Set by An Incremental Learning Model”, in: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2021, Kraków, Poland, June 28 – July 1, 2021, pp. 169–176. https://doi.org/10.1109/CEC45853.2021.9504996

[66] S. Zheng and R. Feng, “A variable projection method for the general radial basis function neural network”, Applied Mathematics and Computation, vol. 451, 128009, 2023. https://doi.org/10.1016/j.amc.2023.128009

[67] Z. Hu, Yu.A. Ushenko, I.V. Soltys, O.V. Dubolazov, M.P. Gorsky, O.V. Olar, and L.Yu. Trifonyuk, Digital Image Processing and Analysis Techniques, Springer Singapore, Singapore, 2024, 115 p. https://doi.org/10.1007/978-981-99-8228-8

[68] Y. Shmelov, S. Vladov, Y. Klimova, and M. Kirukhina, “Expert system for identification of the technical state of the aircraft engine TV3-117 in flight modes”, in: Proceedings of the System Analysis & Intelligent Computing: IEEE First International Conference on System Analysis & Intelligent Computing (SAIC), Kyiv, Ukraine, 08–12 October 2018, pp. 77–82. https://doi.org/10.1109/SAIC.2018.8516864

[69] Cherrat, E. M., Alaoui, R., & Bouzahir, H. (2020). Score Fusion of Finger Vein and Face for Human Recognition based on Convolutional Neural Network Model. International Journal of Computing, 19(1), 11-19. https://doi.org/10.47839/ijc.19.1.1688

Authors’ Profiles

Serhii Vladov is the head of the department of scientific work organization and gender issues at Kremenchuk Flight College of Kharkiv National University of Internal Affairs. He is a candidate for technical sciences. His current research and fields of specialization include the development of applied intelligent systems (including intelligent automatic control systems) for complex dynamic objects, in particular helicopter turboshaft engines. He has nearly 200 scientific papers, among them more than 30 that are indexed in the Scopus scientific database.

Ruslan Yakovliev is the director of Kremenchuk Flight College at the Kharkiv National University of Internal Affairs. He is the organizer of flight and engineering personnel training for Ukraine's Ministry of Internal Affairs units. He is the organizer of flight and engineering personnel training for Ukraine's Ministry of Internal Affairs units. His research focuses on enhancing helicopter safety by implementing intelligent information technologies.

Victoria Vysotska is a postdoctoral researcher at Osnabrück University and an associate professor at the Information Systems and Networks Department of Lviv Polytechnic National University. She received her PhD degree in information technologies from Lviv Polytechnic National University in 2014. She is a Doctor of Technical Sciences and an associate professor. She has currently published more than 450 publications. Her main research interests are focused on NLP, computer linguistics, data science, system analysis, information technologies, and machine learning.

Dmytro Uhryn graduated from Yuriy Fedkovych Chernivtsi National University, Chernivtsi. Currently, he is a Doctor of Technical Sciences and an associate professor at Yuriy Fedkovych Chernivtsi National University. At present, he has authored over 140 publications. His research interests are data mining, information technologies for decision support, swarm intelligence systems, and industry-specific geographic information systems. His research interests include decision-support information technologies, swarm intelligence systems, and industry geoinformation systems.

Artem Karachevtsev, M.Sc. in Electronic Devices (2008). M.Sc. in Computer Science (2023). PhD in Optics and Laser Physics (2013), Vlokh Institute of Physical Optics. Current position: Assistant Professor, Computer Science Department, Yuriy Fedkovych Chernivtsi National University, Ukraine. Research Interests: Computer Science, Software Engineering, Web Development, Biomedical Optics.

How to cite this paper: Serhii Vladov, Ruslan Yakovliev, Victoria Vysotska, Dmytro Uhryn, Artem Karachevtsev, "Polymorphic Radial Basis Functions Neural Network", International Journal of Intelligent Systems and Applications(IJISA), Vol.16, No.4, pp.1-21, 2024. DOI:10.5815/ijisa.2024.04.01