DOI: 10.15587/978-617-8360-23-8.CH3

3 DEVELOPMENT OF THE SCIENTIFIC AND METHODOLOGICAL FRAMEWORK FOR THE INTELLECTUAL ASSESSMENT OF PARAMETERS IN DECISION SUPPORT SYSTEMS

Andrii Shyshatskyi, Ganna Plekhova, Danylo Pliekhov, Oleksii Nalapko, Yuliia Vakulenko, Andrii Lebedynskyi

ABSTRACT

The object of the research is decision support systems. The subject of the research is the process of evaluating parameters in decision support systems. The study proposes: a methodology for the intellectual assessment of parameters in decision support systems and a method for multi-criteria evaluation of hierarchical systems. The originality of the research lies in the use of additional advanced procedures that allow:

– verification of the topology and parameters of decision support systems, taking into account the degree of uncertainty in the input data regarding the known information, achieved through the use of an enhanced penguin swarm algorithm. This reduces the time required for the initial configuration of the evaluation methodology during its setup;

– initial selection of individuals for configuring the evolving artificial neural network, carried out using an enhanced genetic algorithm, which reduces solution search time and increases the reliability of the obtained results;

– exploration of solution spaces for the problem of parameter evaluation in decision support systems, which are described by atypical functions, using the enhanced penguin swarm algorithm;

– configuration of the weights of the evolving artificial neural network improves the accuracy of parameter evaluation in decision support systems;

– utilization of additional mechanisms for correcting the parameters of the evolving artificial neural network is applied by a procedure for modifying the membership function;

– reliability of parameter evaluation in decision support systems is increased by parallel evaluation through multiple assessment methods;

– use of hybrid parameter evaluation for decision support systems allows for correct operation in the absence of conditions of stationarity, homogeneity, normality, and independence.

An example of the application of the proposed scientific and methodological framework for evaluating parameters in decision support systems showed a 25% increase in evaluation reliability, achieved by utilizing additional procedures while maintaining the required operational efficiency.

KEYWORDS

Artificial neural networks, enhanced genetic algorithm, destabilizing factors, metaheuristic algorithm.

The problem of improving the reliability of parameter evaluation in decision support systems is becoming increasingly urgent in modern information systems with various functional purposes [1]. The experience from recent conflicts, involving the use of modern information systems, shows that existing approaches to evaluating parameters in decision support systems do not allow for reliable assessments with the required operational speed [2].

This issue is linked to the following reasons:

– the significant role of the human factor in the evaluation process of decision support system parameters [3];

– the large number of diverse components in decision support systems [3];

– parameter evaluation in decision support systems occurs under conditions of uncertainty, which causes delays in their processing [4];

– the presence of many destabilizing factors that affect the reliability of the parameter evaluation in decision support systems;

– the presence of both structured and unstructured data in decision support systems that need to be processed, among other factors.

Given the diversity, numerous destabilizing factors, and the various dimensions of the indicators describing them, the need for evaluating parameters in decision support systems prompts the search for new approaches. One such approach is the use of metaheuristic algorithms [5–8].

The use of metaheuristic algorithms in their canonical form can improve the operational speed of parameter evaluation in decision support systems. However, further increasing the operational speed of parameter evaluation leads to a deterioration in the reliability of parameter assessments.

This motivates the introduction of various strategies to improve the convergence speed and accuracy of basic metaheuristic algorithms when evaluating parameters in decision support systems. One approach to improving the reliability of parameter evaluation is its further enhancement by combining, comparing, and developing new procedures for their joint use.

An analysis of the works [9–71] shows that common shortcomings in the aforementioned research include:

– the lack of a hierarchical system of indicators for comprehensive evaluation of decision support systems;

– the failure to account for the computational resources of the system managing the evaluation process of decision support system parameters;

– the absence of mechanisms for adjusting the indicator system managing the evaluation process of decision support system parameters;

– the lack of selective engagement of artificial neural network training methods;

– high computational complexity;

– the failure to account for computational (hardware) resources available in the system;

– the absence of prioritized search in a specific direction.

The aim of this research is to develop a methodology for the intellectual assessment of parameters in decision support systems. This will improve the reliability of parameter evaluation in decision support systems with the required operational speed and the generation of subsequent management decisions based on intellectual evaluation.

This will enable the development (or improvement) of software for the operation of decision support systems.

To achieve this aim, the following tasks were set:

– define the algorithm for implementing the methodology;

– provide an example of applying the methodology for the intellectual evaluation of parameters in decision support systems;

– offer recommendations for integrating the proposed methodology into decision support systems.

The object of research is decision support systems. The problem addressed in the research is improving the reliability of parameter evaluation in decision support systems while ensuring the required operational speed, regardless of the volume of incoming data. The subject of research is the process of evaluating parameters in decision support systems.

Parameters in the evaluation system of decision support systems generally include various types of origin and units of measurement, as well as varying degrees of impact on the overall evaluation result. To address this, it is appropriate to use artificial intelligence theory, specifically:

– an enhanced genetic algorithm, which allows automating the evaluation process and conducting random, ordered changes to information and the rearrangement of individuals in the parameter evaluation plane of decision support systems. This enhanced genetic algorithm is also used in this research for the preliminary selection of individuals to improve the reliability of parameter evaluation in decision support systems. The enhanced genetic algorithm is also used for tuning the parameters of an evolving artificial convolutional neural network;

– an enhanced penguin swarm algorithm – for verifying the topology and parameters of decision support systems, as well as the topology and parameters of destabilizing influencing factors. This leads to an increase in the reliability of the obtained parameter evaluation in decision support systems;

– evolving artificial neural networks – enabling the generalized evaluation of decision support system parameters, which are of different origins and units of measurement, taking into account the number of input parameters to be assessed.

The hypothesis of the research is that the reliability of parameter evaluation in decision support systems can be improved with the required operational speed using the intellectual evaluation methodology.

The modeling of the proposed methodology was carried out in the Microsoft Visual Studio 2022 programming environment (USA). The task solved in the simulation process was the determination of the composition of a military grouping (forces). The hardware used in the research process is the AMD Ryzen 5.

Parameters for the operation of the enhanced algorithm:

– number of iterations – 25;

– number of individuals in the swarm algorithm – 25;

– range of feature space – [–100, 100].

The structure of the evolving artificial neural network is presented in the work [20].

3.1 ALGORITHM FOR IMPLEMENTING THE INTELLECTUAL PARAMETER EVALUATION METHODOLOGY IN DECISION SUPPORT SYSTEMS

The methodology for the intellectual evaluation of parameters in decision support systems consists of the following sequence of actions:

Action 1. Input of initial data.

At this stage, the available initial data on decision support systems and destabilizing influencing factors are input, specifically:

– the number and type of technical means included in the decision support systems;

– the number and type of destabilizing factors that affect the objectivity of the evaluation of the state of the decision support systems;

– technical characteristics of the means included in the decision support systems;

– technical characteristics of destabilizing factors that affect the objectivity of the evaluation of the state of decision support systems;

– topology of connections within the decision support systems;

– topology of connections of destabilizing factors;

– the type of data circulating within decision support systems;

– available computational resources of the decision support systems;

– information about the operational environment of the decision support systems, etc.

This procedure involves the processing of arrays at the initial observation window, exponential normalization of the data, and setting tasks for the learning, testing, and forecasting processes.

Action 2. Verification of parameters necessary for calculations.

At this stage, the initial data about the decision support system and destabilizing factors are clarified. This is done by taking into account the type of uncertainty about the state of decision support systems using the enhanced penguin swarm algorithm proposed by the authors in work [20].

Action 3. Formation of the topology of the evolving artificial neural network.

At this stage, the enhanced penguin swarm algorithm is used to form the topology of the evolving artificial neural network, proposed by the authors in work [20], based on the verified data.

Action 4. Preliminary selection of individuals for the genetic algorithm.

To improve the reliability of the obtained solutions, the preliminary selection of individuals is carried out using the enhanced genetic algorithm proposed by the authors in study [19]. The enhanced genetic algorithm is further used in *Action 5.3*.

Action 5. Parallel evaluation of the decision support system's state using multiple approaches.

Action 5.1. Evaluation of the decision support system's state based on the multiple regression algorithm.

The traditional technology for sequential evaluation and forecasting the state of decision support systems based on observations containing a stochastic component relies on the mathematical apparatus of multivariate regression [13].

In general, multivariate regression generalizes the one-dimensional linear regression algorithm to the situation of multiple interdependent variables *X*, which define the structure of the base model.

The multiple regression algorithm includes the description of the dependence of the predicted parameters on the values of the input parameters, i.e., the regressors, which are the control parameters of the decision support system.

For linear forecasting, the application of regression analysis is based on the possibility of sequentially evaluating the state parameters, control, and output parameters of decision support systems.

Let's assume that the average values of the predicted output characteristics of the decision support systems $Z_{k+\tau} = \left(z_1, \ldots, z_{M_y}\right)_{k+\tau}$, $k = 1, \ldots, N$ are related to the state parameters, which include control parameters $X_k = \left(x_1, \ldots, x_{M_X}\right)_k$, $k = 1, \ldots, N$ in a functional dependency of the form

$$Z_{k+\tau} = f\left(X_k\right) + V_k,\ k = 1, \ldots, N. \tag{3.1}$$

It is assumed that the additive noise (in our case, intentional destructive influence) is centered $EV_k = 0$, $k = 1, \ldots, N$.

The task of regression evaluation is to establish the form of the relationship between dependent and independent variables over time. For the task of corrective control, the functional dependency (3.1) allows for linearization, which makes it possible to restrict the model to linear regression

$$Z_{k+\tau} = C_k X_k + V_k,\ k = 1, \ldots, N. \tag{3.2}$$

The rapid aging of data, caused by the transient nature of military grouping (force) operations, formed by a non-stationary process, results in the use of a multidimensional sample on a sliding observation window of size L as the initial data. In this case, the output data arrays at each forecasting step are specified by matrices

$$X_{L:M_X} = \begin{bmatrix} x_{11}, & x_{12} & \ldots & x_{1M_x} \\ x_{21}, & x_{22} & \ldots & x_{2M_x} \\ \ldots & \ldots & \ldots & \ldots \\ x_{L1}, & x_{L2} & \ldots & x_{LM_x} \end{bmatrix};\ Z_{L:M_Y} = \begin{bmatrix} z_{11}, & z_{12} & \ldots & z_{1M_y} \\ z_{21}, & z_{22} & \ldots & z_{2M_y} \\ \ldots & \ldots & \ldots & \ldots \\ z_{L-\tau,1}, & z_{L-\tau,2} & \ldots & z_{L-\tau,M_y} \end{bmatrix}.$$

Then, based on the minimization of the quadratic functional

$$V^T V = \left(Z - XC\right)^T \left(Z - XC\right) = Z^T Z - 2C^T X^T Z + C^T X^T XC,$$

i.e., using the least squares method, it is possible to obtain the well-known matrix expression for the predictive transfer coefficient of linear regression

$$C = \left(X^T X\right)^{-1} X^T. \tag{3.3}$$

In this case, the linear regression forecasting algorithm is described by the simplest matrix relationship of the form

$$Z_{k+\tau} = C_k X_k, \tag{3.4}$$

where $Z_{k+\tau} = \left(z_1, \ldots, z_{M_Z}\right)^T_{k+\tau}$, the regressors use only those state parameters of the decision support system (decision support system) components that allow manipulation of values during the control process, i.e., control parameters $U_k = \left(u_1, \ldots, u_{M_U}\right)^T_k$.

The traditional linear regression scheme includes important assumptions known as the Gauss-Markov conditions [13, 14]. This algorithm fits the requirements for adaptation, which is associated with the change in the coefficient of pairwise correlation.

The second feature of the developed algorithm is the application of a sliding observation window. It is important to note that within this window, the output array of forecasted parameter values for decision support systems $Z_{L:M_Y}$ should be shifted backwards by τ time steps relative to the array of regressors $X_{L:M_X}$.

The main forecasting cycle occurs over the sliding observation interval. At each step, the current mean values and covariance structures are corrected.

The forecast of the parameter values for decision support systems is performed by the previously described method of vector multiplication of the current centered monitoring data values and the matrix transfer coefficient of the least squares filter. The justification for the optimality of this approach directly follows from the well-known Gauss-Markov theorem [9].

The values of the parameters for the decision support system at the output of the predictor form the vector of the evolution of the output parameters. Typically, the quality indicators used are the root mean square deviation (RMSD) of the forecast or the average values of the obtained errors, which allows for a forecast that is quite close to the actual process trajectory (the average relative error does not exceed 9%).

Action 5.2. Evaluation of the decision support system's state based on the enhanced canonical correlation method.

The enhanced canonical correlation method is a generalization of multiple correlation for the case where there are two or more interrelated variables *X* and *Y* [9, 18]. From the perspective of forming a linear forecast, the application of canonical correlations means the ability to simultaneously evaluate a group of interrelated output parameters, considered as generalized linear combinations of interrelated parameters. Let's consider the mathematical apparatus of canonical correlations. Let's define possible linear combinations for *q* variables *Y* and *p* variables *X* in the general population

$$X^* = \sum_{i=1}^{p} \alpha_i X_i;\ Y^* = \sum_{j=1}^{q} \beta_j Y_j.$$

The tasks of canonical correlations include determining the coefficients α_i and β_i [10].

Let's consider the algorithm for multivariate analysis based on the enhanced canonical correlation method, where the output array is divided into observed and unobserved parts: $X \in N_p\{\mu_1, P_1\}$ and $Y \in N_q\{\mu_2, P_2\}$, respectively. In this case, the covariance matrix will take the following form

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}. \tag{3.5}$$

Then, the correlation coefficient will be

$$\rho_c = \frac{\operatorname{cov}(X\alpha, Y\beta)}{\sqrt{\operatorname{var}(X\alpha)\cdot\operatorname{var}(Y\beta)}} = \frac{\alpha^T P_{12}\beta}{\sqrt{(\alpha^T P_{11}\alpha)\cdot(\beta^T P_{22}\beta)}}.$$

Now, let's assume that the state of the decision support system at time k is described (with sufficient accuracy) by m parameters, combined into a vector $X = (x_1, \dots, x_m)$. Geometrically, this means that the state of the decision support system is a point in the m-dimensional phase space R^m. Let there be measurement results for these parameters at time moments $k = 1, \dots, N$.

Let's combine the obtained measurement results of the decision support system parameters into a matrix X of size $n\times m$. The row of this matrix with index i corresponds to the result of the i-th vector measurement, $i = 1, \dots, n$, while the column with index j represents the set of n measurement values of the j-th parameter of the decision support system in each measurement, $j = 1, \dots, m$.

The task is to refine or create a mathematical model based on the available data that fits the tasks of both forecasting parameter values and control. In the case of non-stationary processes, the model is not universal and requires constant reconfiguration.

If the data is normalized, estimates of the covariance and correlation matrices are constructed. The second estimate (the correlation matrix) involves normalizing the standard deviation; if this operation has already been performed, the results of these estimates coincide. These calculations are carried out for known values of α and P, which are replaced by their estimates. The quality of the estimate will depend on the size and reliability of the initial data on the parameters of the decision support system.

Consider the following task: there are m parameters, of which p are observable, and the remaining q are unobservable. The task is to estimate the predicted parameters based on the available output data.

As an assumption, let's consider that the initial data is already normalized and centered. The covariance matrix will take the form (3.5), where P_{11} – the covariance matrix of the observed parameters, which values are obtained during monitoring, P_{22} – the covariance matrix of the predicted parameters, and P_{12} – the cross-covariance matrix between the observed and unobserved parameters.

The task is to find the matrix of weighting coefficients C under the condition of minimizing the average sum of squared residuals

$$trE\left[(x_2 - C\cdot x_1)^T (x_2 - C\cdot x_1)\right] \to \min.$$

Let's transform this expression into the form

$$trE\left[\left(x_2-C\cdot x_1\right)^T\left(x_2-C\cdot x_1\right)\right]=$$
$$=tr\left[E\left(x_2x_2^T\right)-C\cdot E\left(x_1x_2^T\right)-E\left(x_2x_1^T\right)\cdot C^T+C\cdot E\left(x_1x_1^T\right)\cdot C^T\right]=$$
$$=tr\left[P_{22}-C\cdot P_{12}-P_{21}^T\cdot C^T+C\cdot P_{11}\cdot C^T\right]\to \min.$$

By substituting the initial data, the formula for the optimal linear estimate of the vector X_2 based on the known vector X_1 will take the form

$$\hat{X}_2=E\left(X_2\right)+P_{12}^T\cdot P_{11}^{-1}\cdot\left(X_1-E\left(X_1\right)\right). \tag{3.6}$$

Now, using sample estimates

$$\hat{X}_2=\overline{X}_2+\hat{P}_{12}^T\cdot\hat{P}_{11}^{-1}\cdot\left(X_1-\overline{X}_1\right).$$

In terms of the task of forecasting the output characteristics of the decision support system, expression (3.7) will take the form

$$\tilde{Z}_{t+\tau}=\overline{Z}_{t-L,t}+\hat{P}_{UZ}^T\cdot\hat{P}_u^{-1}\cdot\left(U_t-\overline{U}_{t-L,t}\right). \tag{3.7}$$

Substituting the found value of the matrix into the expression for the average sum of squared residuals, the covariance matrix of the estimation errors will take the form

$$P_v=P_{22}-2C\cdot P_{12}+C\cdot P_{11}\cdot C^T=P_{22}-2P_{12}^T\cdot P_{11}^{-1}\cdot P_{12}+P_{12}^T\cdot P_{11}^{-1}\cdot P_{11}\cdot P_{11}^{-1}\cdot P_{12}=$$
$$=P_{22}-P_{12}^T\cdot P_{11}^{-1}\cdot P_{12}. \tag{3.8}$$

The diagonal elements of this matrix represent the variance of the estimates of the corresponding components (in dimensionless units). Typically, confidence intervals calculated using this formula turn out to be overly pessimistic. The variances can be determined by calculating the sum of the squared errors in the prediction based on the available dataset, but they will only be relevant to this specific data. The connection of these variances with the corresponding theoretical characteristics depends on the size of the available sample of parameters for the decision support system.

At the same time, training is carried out based on data from a sliding observation window.

Action 5.3. Hybrid parameter estimation for the decision support system.

As mentioned earlier, the statistical algorithms proposed above for parameter estimation in decision support systems provide the best solution when a number of assumptions (stationarity, homogeneity, normality, independence, etc.) are met, which, in practice, are not always valid.

However, a complete rejection of statistical algorithms for forecasting the parameters of decision support systems is also irrational. The universality of the quadratic criterion allows obtaining good initial approximations to the averaged dynamics of the forecasted process. Hence, the study proposes the development of a hybrid algorithm that combines multivariate statistical analysis algorithms with a computational scheme that self-develops, based on evolutionary modeling methods. The core idea is to replace the optimization of a dynamic system with its evolutionary process. In fact, this refers to the stochastic self-organization of the applied mathematical model.

Let's assume that, based on the traditional statistical algorithm $A\left\{S\left(A\right),x\right\}$, characterized by a given structure $S(A)$ and a set of parameters x, the necessary output parameter of the decision support system $\hat{y}$ is estimated.

In this case, the effectiveness of the algorithm $Eff(A)$ is evaluated based on its application to the output data from the sliding window. The effectiveness indicator is typically represented by the general quality metrics described earlier or local accuracy measures, such as the total square of the prediction error.

At this stage, the second part of the enhanced genetic algorithm, proposed in work [19], is used. Let's introduce two operators: the variability operator $Var(A): A \Rightarrow \left\{A_1,\ldots,A_{N_a} : A_i \neq A_j \neq A, \forall i,j\right\}$ and the selection operator $Sel\left(A_1,\ldots,A_{N_g}\right):\left\{A_1,\ldots,A_{N_g}\right\} \Rightarrow \left\{A_{<1>},\ldots,A_{<N_a>} : Eff\left(A_{<1>}\right)\geq\ldots\geq Eff\left(A_{<N_a>}\right)\geq Eff\left(A_j\right), \forall j > N_a\right\}$, where N_a – the number of "selected" algorithms that are used for further reproduction; $N_g=N_a(1+ N_d)$ – the number of strategies for one generation that are subject to selection, N_d – the number of strategy offspring generated according to the specified rules at each iteration.

Let $A_0=A\{S_0(x), x_0\}$ – be a certain variant of the forecasting algorithm with given parameters and structure, accepted as the baseline "father" algorithm. Then, the technology of evolutionary modeling reduces to the repeated application of the sequence of operators

$$\begin{array}{c} A_o \Rightarrow Var\left(A_0\right)=\left\{A_a\right\}=\left\{A_1,\ldots,A_{N_g}\right\} \Rightarrow Var\left(A_d\right)=\left\{A_d\right\}=\left\{A_1,\ldots,A_{N_d}\right\} \\ \Uparrow \qquad\qquad\qquad\qquad\qquad \Downarrow \\ \psi\left(A_1,\ldots,A_{N_g}\right)=A_0=\left\{A_{<1>},\ldots,A_{<N_a>}\right\} \Leftarrow \left\{A_g\right\}=\left\{A_a \cup A_d\right\}. \end{array} \tag{3.9}$$

The presented approach to evolutionary optimization, combined with the previously described algorithm based on the canonical correlation method, forms a unified hybrid algorithm. This algorithm retains all the advantages of statistical analysis and supplements them, allowing for the avoidance of the drawbacks associated with the lack of Gaussianity and stationarity in real observation series for the parameters of non-stationary complex technical objects.

Action 6. Formation of a generalized parameter evaluation for the decision support system.

Based on the evolving artificial neural network, a generalized evaluation of the state of the decision support system is formed. This is done through the convolution of each group of parameters for the system's state. The architecture of the evolving artificial neural network for evaluating the parameters of the decision support system is presented in work [20].

Action 7. Verification of the stop criterion for the combined algorithm.

The algorithm terminates if the maximum number of iterations has been reached. Otherwise, the generation of new positions and checking of conditions are repeated.

Action 8. Determining the number of required computational resources for evaluation.

To avoid the cyclic repetition of calculations in *Actions 1–8* of this method and to increase computational efficiency, the system's load is additionally determined. If the computational complexity exceeds the established threshold, the number of software-hardware resources that need to be added is determined using the method proposed in work [20].

Action 9. Training the knowledge bases of agents. At this stage, the training of the knowledge bases of agents from the list of bio-inspired algorithms used in this study is performed. The method of deep learning proposed in work [20] is used as the learning method.

End.

3.2 EXAMPLE OF APPLYING THE PROPOSED METHODOLOGY FOR PARAMETER EVALUATION IN DECISION SUPPORT SYSTEMS

The evaluation of the effectiveness of the proposed methodology for parameter evaluation in decision support systems, based on their own quality indicators, such as the root mean square deviation (3.7), the mean value of the sum of squared residuals (3.8), and other similar characteristics, allows comparing them by the degree of reliability of the evaluation and forecasting.

However, it does not provide answers to questions about the advisability of improving the values of these indicators. Like any mathematical or informational tool, the effectiveness of the proposed methodology for evaluation and forecasting can only be assessed through the quality indicators of the metasystem for which it was created and improved.

In this context, the metasystem is represented by the proactive decision support system. The performance indicators of such a system are the external or exogenous numerical characteristics that are hierarchically specified by the higher-level decision support system.

The suitability criterion for the forecasting algorithm (3.9) is the verification of the condition for the membership of the forecasted values of the parameters in the state vector of the decision support system within the constraint set $\left\{\left|x_i^* \pm \Delta_i\right| \wedge \Omega_{per}^i\right\}$, $\forall i = 1,\ldots,M$.

Here $x_i^* \pm \Delta_i$, $\forall i = 1,\ldots,M$ – the sets of constraints that correspond to the requirement for stabilizing the values of the decision support system's parameters around a reference value x_i^*, $\forall i = 1,\ldots,M$, which is determined by the regulation of the controlled parameter in the decision support system, Ω_{per} – the set of technical constraints imposed on the parameters of the decision support system.

As an example, let's build a proactive parameter evaluation system for the decision support system based on the algorithm for iterating through the possible values of control parameters.

The formation of the ε-neighborhood $\Delta = \Delta\left(U_0\left(t\right)\right)$ can be carried out using several methods:

1) $\Delta = \left[U_0 - R/2; U_0 + R/2\right]$, where $R = abs\left(U_{\max} - U_{\min}\right)$; $U_{\max}$, $U_{\min}$ – the boundaries of the acceptable range of the control parameter changes in the decision support system;

2) $\Delta = \left[U_0 - s\left(U\right); U_0 + s\left(U\right)\right]$, where $s\left(U\right)$ – the standard deviation of the change in the decision support system's parameter;

3) $\Delta = \left[U_0 - t_0 * s / sqrt(N); U_0 + t_0 * s / sqrt(N)\right]$, where t_0 – the critical value of the t-statistic for the Student's distribution at the chosen confidence level α, N – the number of observations in the sliding window;

4) $\Delta = \left[U_0 - \%R * U_0; U_0 + \%R * U_0\right]$, where $\%R$ – the half-interval used to search for the best solution (for example, $\%R$=0.05 for a 5%-th deviation).

Next, the number of steps for iterating within the parameter change range is established N_{step} (e.g., 10). The total number of possible variants of the parameter evaluations for the decision support system, formed as the number of permutations with repetition, is given by $\left(N_{step}\right)^{M_{man}}$, where M_{man} – the number of parameters in the decision support system that are used for manipulation.

It is important to note that the number of possible controls grows rapidly with an increase in N_{step} and M_{man}. Examples of the number of possible evaluation variants for different values of N_{step} and M_{man} are presented in **Table 3.1**.

Table 3.1 Number of iterations for parameter evaluation variants in the decision support system

M_{man}	N_{step}				
	5	10	15	20	25
2	25	100	225	400	625
3	125	1000	3375	8000	15625
4	625	10000	50625	160000	390625
5	3125	100000	759375	3200000	9765625

Considering that each step is associated with a considerable number of operations, including the calculation of inverse matrices, the growth of these parameters should be done while taking into account the computational capabilities of the hardware. According to the adopted algorithm, for each variant of parameter evaluation, a forecast is formed based on regression, neural network, or other technologies. Comparing the forecasted output parameter values against each other, considering the set of technological constraints imposed on the parameters of the decision support system, allows for the direct identification of the optimal value of the state parameter of the decision support system at a given time.

Comparing the obtained result with the traditional scheme of situational assessment, which is implemented during the management of the decision support system, allows for the evaluation of the terminal effectiveness of hybrid evaluation and forecasting through the quality indicators of the higher-level system for which it was created.

3.3 RECOMMENDATIONS FOR INTEGRATING THE PROPOSED METHODOLOGY INTO DECISION SUPPORT SYSTEMS

As an example of implementing hybrid evaluation and forecasting of decision support system parameters for non-stationary processes, let's consider the option of building it based on the back estimation (BE) procedure of the state parameters of the decision support system. The formation of improved state evaluation of the decision support system is carried out by sequential (step-by-step) modification of the chosen initial output parameter and the back-calculation of the output parameters (with improved outcomes) into manipulation parameters (control parameters used in the current situation).

Let's consider the formalized formulation of intellectual evaluation and forecasting of parameter values for the decision support system based on the development of an algorithm for back estimation of possible parameter values.

There is an initial value of the parameter U_0, obtained from the data about the decision support system under consideration. Then, using a predefined improvement step δY for the system's state indicator $Y = Y_0 + \delta Y$, the evaluation effectiveness according to the defined evaluation criterion is found to be higher, i.e., $Eff\left(Y\right) > Eff\left(Y_0\right)$.

The step size is selected taking into account the physical and technical characteristics of the specific decision support system. In the considered example, as already mentioned, it was chosen as 2–3% of the forecast, estimated based on the current state of the decision support system. Using data from a limited sliding window, as described earlier, the parameters of the intellectual parameter evaluation methodology are refined.

This methodology is based on generalized linear regression and the linking parameter that relates the state evaluation parameter to its output parameters and state parameters $\tilde{Y}_{k+1} = F\left(U_k\right),\ k = L+1,\ldots,N$. For a non-degenerate operator F it is possible to construct an inverse mapping $\widehat{U}_k^* = F^{-1}\left(\tilde{Y}_{k+1} + \delta Y_{k+1}\right)$, which allows obtaining the values of the control parameters $\widehat{U}_k^*$, that have increased efficiency compared to the reference control being compared. At the same time, it is necessary to additionally verify the condition of admissibility for the found control values $\widehat{U}_k^*$ and other state parameters of the decision support system $\widehat{X}_k^*$, i.e., the membership of the corresponding numerical values of the parameters in the set of admissible values (3.11).

The consideration of the variation of state parameters in the decision support system is carried out by using a sliding observation window. The size of the window is selected based on the dynamics of variation in the average values of the controlled parameters.

The core of neural network-based forecasting technologies is the iterative refinement of the weight coefficients of the multiplicative inputs of nonlinear nodes, unified by a single network structure [19, 20].

The process of correcting weight coefficients is carried out according to the feedback signal, formed by the difference between the network's output signals and the actual measured values, combined with the corresponding input signals into the training dataset. Let's consider an example of evaluating and forecasting the state of the decision support system.

Let $\left(x_1, x_2, \ldots, x_p\right)^T$ – be the input parameters, $w^1 = \left(w_{11}^1, w_{22}^1, \ldots, w_{p1}^1\right)^T$, $w^{12} = \left(w_{11}^2, w_{22}^2, \ldots, w_{p1}^2\right)^T$ – be the boosting coefficients of the first and second generations of the models. The artificial neural network, with an evolving structure, has a different number of neurons at each level: level A (input layer) – p neurons, level S (first layer) – l neurons, and level R (second layer) – k neurons.

Let N be the number of input and output points obtained during the experiment or through simulation, $X = \left(X_1, X_2, \ldots, X_p\right)$ – be the input vector, $D = \left(d_1, d_2, \ldots, d_k\right)$ – be the real or computed outputs [15, 18].

The objective function to be minimized is as follows

$$E(\varpi)=\frac{1}{2}\left[\sum_{i=1}^{N}\sum_{j=1}^{l}\left(y_{ij}^{1}-d_{ij}^{1}\right)+\sum_{i=1}^{N}\sum_{j=1}^{k}\left(y_{ij}^{2}-d_{ij}^{2}\right)\right]^{2}=\frac{1}{2}\sum_{i=1}^{N}\left(\sum_{j=1}^{l}\left(y_{ij}^{1}-d_{ij}^{1}\right)+\sum_{j=1}^{k}\left(y_{ij}^{2}-d_{ij}^{2}\right)\right)^{2}.$$

Minimization is achieved through gradient descent, meaning the adjustment of weight coefficients is formulated as

$$\Delta\varpi_{ij}^{(n)}=-\eta\frac{\partial E}{\partial\omega_{ij}},\quad n=1,2,$$

where $\varpi_{ij}^{(n)}$ – the weight characteristic of the connection between the *i*-th neuron of the *n*-1 level and the *j*-th neuron of the *n*-th level is *n*, $1<0<\eta$ – the learning rate coefficient. It is known that $\frac{\partial E}{\partial\omega_{ij}}=\frac{\partial E}{\partial y_j}\cdot\frac{\partial y_j}{\partial S_j}\cdot\frac{\partial S_j}{\partial\omega_{ij}}$, where y_j – the output of the neuron, $S_j=\sum_{i=1}^{N}\omega_{ij}x_{ij}$ – the weighted sum of its input signals (the argument of the activation function). Typical activation functions used are the sigmoid $A=\frac{1}{1+e^{-\sum\omega_i x_i}}$ or the hyperbolic tangent $A=\operatorname{th}x=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$, $\operatorname{th}'x=\frac{1}{ch^{2}(x)}$, where $ch(x)$ – the hyperbolic cosine, th *x* is the hyperbolic tangent, $\left(\operatorname{th}x\right)'=1-\left(\operatorname{th}x\right)^{2}$. The third factor represents the output of the previous layer neuron. The first factor can be expanded as $\frac{\partial E}{\partial y_i}=\sum_k\frac{\partial E}{\partial y_k}\cdot\frac{\partial y_k}{\partial S_k}\cdot\frac{\partial S_k}{\partial y_i}=\sum_k\frac{\partial E}{\partial y_k}\cdot\frac{\partial y_k}{\partial S_k}$.

The last sum is calculated over the neurons of the (*n-1*)-th layer. Let's introduce a new substitution: $\delta_j^{(n)}=\frac{\partial E}{\partial y_j}\cdot\frac{\partial y_j}{\partial S_j}$ which gives the recursive formula $\delta_j^{(n)}=\left[\sum_k\delta_k^{(n+1)}\cdot\omega_{jk}^{(n+1)}\right]\cdot\frac{\partial y_j}{\partial S_j}$, which allows, knowing $\delta_j^{(n+1)}$, to compute $\delta_j^{(n)}$. For the output layer $\delta_e^{(n)}=\left(y_e^{(n)}-d_e\right)\cdot\frac{dy_e}{dS_e}$. Then, the weight adjustment will have the form $\Delta\varpi_{ij}^{(n)}=-\eta\cdot\delta_j^{(n)}y_i^{(n-1)}$, $n=1,2$ [5, 8].

To give the weight correction process some inertia, in order to smooth out abrupt jumps when moving across the surface of the objective function, the last expression is supplemented with the values of the weight changes from the previous iteration

$$\Delta\varpi_{ij}^{(n)}=-\eta\cdot,\left(\mu\cdot\Delta\omega_j^{(n)}\cdot\left(t-1\right)+\left(1-\mu\right)\cdot\delta_j^{(n)}y_i^{(n-1)}\right),\ n=1,2,$$

where μ – the inertia coefficient, *t* – the current iteration number.

For the sigmoid function $\delta_e^{(n)}=\left(y_e^{(n)}-d_e\right)\cdot\left(1-S_e\right)\cdot S_e$, and for the hyperbolic tangent function $\delta_e^{(n)}=\left(y_e^{(n)}-d_e\right)\cdot\left(1-S_e^{2}\right)\cdot S_e$ [5, 8].

Let's evaluate the effectiveness of the proposed parameter evaluation methodology for the decision support system in comparison with the known approaches for evaluating such systems. The results of the evaluation based on the reliability of the decisions made are presented in **Table 3.2**.

Table 3.2 Evaluation of the effectiveness of the proposed methodology for parameter evaluation in decision support systems

Approach name	Evaluation completeness	Accuracy	Sensitivity	Average value
Densenet 201	0.6163	0.4243	0.4485	0.4335
Densenet 121	0.9523	0.8489	0.8590	0.8588
MobileNetV2	0.9289	0.9295	0.9289	0.9287
DenseNet-SEGR	0.9588	0.9514	0.9511	0.9512
Gradient boosting classifier	0.92021	0.91128	0.9003	0.91449
KNN	0.8736	0.8839	0.88529	0.9003
LSTM	0.7981	0.8005	0.8191	0.8217
RNN	0.8014	0.8122	0.8022	0.8101
CNN	0.9232	0.9104	0.9271	0.9301
Proposed method	0.9511	0.9611	0.9601	0.9612

As seen from **Table 3.2**, the reliability of the parameter evaluation in decision support systems is improved by 17–21% due to the use of additional procedures, while maintaining the required operational speed.

3.4 METHOD OF MULTI-CRITERIA EVALUATION OF HIERARCHICAL SYSTEMS

The method of multi-criteria evaluation of hierarchical systems consists of the following sequence of actions:

Action 1. Input of initial data.

At this stage, the available initial data for the beginning of the multi-criteria evaluation method are entered. The following information is introduced at this stage:

– the number of subsystems in the hierarchical system;

– characteristics of each subsystem in the hierarchical system (the number of elements in each subsystem, the number of connections between each element in the subsystem, the type of element in the subsystem (purpose, main technical characteristics), etc.);

– the number of connections between each subsystem (or a specific element) in the hierarchical system;

– the type and number of individual elements of the hierarchical system that are not part of any subsystem of the hierarchical system.

Action 2. Verification of the entered data and clarification of the relationships between the elements of the hierarchical system.

To reduce the subjectivity of the obtained evaluation, at this stage, the entered data is verified, and the relationships between the elements of the hierarchical system are clarified and described using the enhanced penguin swarm algorithm proposed in work [20].

Action 3. Description of external and internal factors affecting the hierarchical system being analyzed.

At this stage, the list of external factors affecting the functioning process of the hierarchical system is defined, along with their degree of influence on the functioning process of the hierarchical system. Internal factors present within the system are also introduced. This procedure is based on the evaluation method proposed in study [19], which uses the mathematical framework of fuzzy cognitive models.

Action 4. Verification and clarification of the established factors.

The procedure for verifying and clarifying the established factors consists of two stages. In the first stage, it involves using failure tree analysis and the interval-valued fuzzy Pythagorean hierarchical process to rank and select the most critical factors. In the second stage, it involves using the interval-valued fuzzy Pythagorean method for evaluating and visualizing the cause-and-effect relationships between the selected factors.

Action 4.1. Reduction of uncertainty using the interval-valued pythagorean fuzzy set.

In this study, a combination of fuzzy set theory (in this case, the Pythagorean fuzzy set) with multi-criteria evaluation methods is proposed to structure and solve complex decision-making tasks that involve broad and hierarchically organized criteria. This combination is widely used to overcome the inaccuracies that arise when relying on expert evaluation in multi-criteria evaluation methods.

The Pythagorean fuzzy set is defined as follows

$$P=\left\{\left\langle x,P\left(\mu_p(x),\nu_p(x)\right)\right\rangle;x\in X\right\}, \tag{3.10}$$

where X – a finite set, $\mu_p(x):X\mapsto\left[0,1\right]$ and $\nu_p(x):X\mapsto\left[0,1\right]$ – the degree of membership and degree of non-membership of an element $x\in X$ to the set P. The values of $\mu_p(x)$ and $\nu_p(x)$ must satisfy the following conditions

$$0\le\mu_p(x)^2+\nu_p(x)^2\le 1.x\in X. \tag{3.11}$$

The degree of uncertainty of the Pythagorean fuzzy set with respect to the set P can be calculated as follows

$$\pi_p(x)=\sqrt{1-\mu_p(x)^2-\nu_p(x)^2}. \tag{3.12}$$

For a more precise representation of variation and uncertainty, the interval-valued Pythagorean fuzzy set is used, in which intervals are employed to represent the degree of membership instead of point values. The set $\tilde{P}$ is defined as follows

$$\tilde{P}=\left\{\left\langle x,\tilde{P}\left(\left[\mu_{\tilde{P}_L}(x),\mu_{\tilde{P}_U}(x)\right],\left[\nu_{\tilde{P}_L}(x),\nu_{\tilde{P}_U}(x)\right]\right)\right\rangle;x\in X\right\}, \tag{3.13}$$

where $\left\langle\left[\mu_{\tilde{P}_L}(x),\mu_{\tilde{P}_U}(x)\right],\left[\nu_{\tilde{P}_L}(x),\nu_{\tilde{P}_U}(x)\right]\right\rangle$ – the interval-valued Pythagorean fuzzy number $0\le\mu_{\tilde{P}_L}(x)\le$ $\le\mu_{\tilde{P}_U}(x)\le\nu_{\tilde{P}_L}(x)\le\nu_{\tilde{P}_U}(x)$. $\mu_{\tilde{P}_L}(x)$ and $\nu_{\tilde{P}_L}(x)$ must satisfy the expression

$$0\le\mu_{\tilde{P}_L}(x)^2+\nu_{\tilde{P}_L}(x)^2\le 1.x\in X. \tag{3.14}$$

The value of uncertainty for the interval-valued Pythagorean fuzzy set with respect to $\tilde{P}$ can be calculated as follows

$$\pi_{\tilde{P}}(x)=\left[\pi_{\tilde{P}_L}(x),\pi_{\tilde{P}_U}(x)\right]=\left[\sqrt{1-\mu_{\tilde{P}_U}(x)^2-\nu_{\tilde{P}_U}(x)^2},\sqrt{1-\mu_{\tilde{P}_L}(x)^2-\nu_{\tilde{P}_L}(x)^2}\right]. \quad (3.15)$$

Action 4.2. Evaluation and visualization of cause-and-effect relationships between selected factors.

Given that one interval Pythagorean fuzzy set is provided to describe the cause-and-effect relationships between the selected factors, there is the following expression

$\tilde{P}=\left(\left[\mu_{\tilde{P}_L}(x),\mu_{\tilde{P}_U}(x)\right],\left[\nu_{\tilde{P}_L}(x),\nu_{\tilde{P}_U}(x)\right]\right),\left[\mu_{\tilde{P}_L}(x),\mu_{\tilde{P}_U}(x)\right]\subseteq[0,1],\left[\nu_{\tilde{P}_L}(x),\nu_{\tilde{P}_U}(x)\right]\subseteq[0,1],$ and

$0\le\mu_{\tilde{P}_L}(x)^2+\nu_{\tilde{P}_L}(x)^2\le1$, with parameter $\lambda>0$. In this case, the following operation is performed:

$$\lambda\tilde{P}=\left(\left[\sqrt{1-\left(1-\mu_{\tilde{P}_L}(x)^2\right)^{\lambda}},\sqrt{1-\left(1-\mu_{\tilde{P}_U}(x)^2\right)^{\lambda}},\left[\nu_{\tilde{P}_L}(x)^{\lambda},\nu_{\tilde{P}_U}(x)^{\lambda}\right]\right]\right), \quad (3.16)$$

$$\lambda\tilde{P}=\left(\left[\mu_{\tilde{P}_L}(x)^{\lambda},\mu_{\tilde{P}_U}(x)^{\lambda}\right],\left[\sqrt{1-\left(1-\nu_{\tilde{P}_L}(x)^2\right)^{\lambda}},\sqrt{1-\left(1-\nu_{\tilde{P}_U}(x)^2\right)^{\lambda}}\right]\right), \quad (3.17)$$

$\tilde{P}=\left(\left[\mu_{\tilde{P}_L}(x),\mu_{\tilde{P}_U}(x)\right],\left[\nu_{\tilde{P}_L}(x),\nu_{\tilde{P}_U}(x)\right]\right),\left[\mu_{\tilde{P}_L}(x),\mu_{\tilde{P}_U}(x)\right]\subseteq[0,1],\left[\nu_{\tilde{P}_L}(x),\nu_{\tilde{P}_U}(x)\right]\subseteq[0,1],$ and

$0\le\mu_{\tilde{P}_L}(x)^2+\nu_{\tilde{P}_L}(x)^2\le1$.

Given that two interval-valued Pythagorean fuzzy sets $\tilde{P}_1=\left(\left[a_1,b_1\right],\left[c_1,d_1\right]\right)$ and $\tilde{P}_2=\left(\left[a_2,b_2\right],\left[c_2,d_2\right]\right)$ are provided, the following operation is performed:

$$\tilde{P}_1\oplus\tilde{P}_2=\left(\left[\sqrt{a_1^2+a_2^2-a_1^2a_2^2,b_1^2+b_2^2-b_1^2b_2^2}\right],\left[c_1c_2,d_1d_2\right]\right), \quad (3.18)$$

$$\tilde{P}_1\oplus\tilde{P}_2=\left(\left[a_1a_2,b_1b_2\right],\left[\sqrt{c_1^2+c_2^2-c_1^2c_2^2,d_1^2+d_2^2-d_1^2d_2^2}\right]\right). \quad (3.19)$$

Action 5. Vulnerability analysis of the subsystem (individual element) of the hierarchical system.

Fault tree analysis is widely used to identify potential root causes, referred to as basic events, as well as to determine the probability of an unexpected event, known as the top event. The top event is placed at the top of the tree, while the basic events are at the bottom. Basic events (BE) within the fault tree are considered statistically independent and are combined using logical operators (AND/OR).

The fault tree analysis includes both qualitative and quantitative assessments. In the qualitative evaluation, the fault tree establishes and explains the theoretical relationships between the fault tree and basic events based on "AND" and "OR" logic. In the quantitative assessment, the basic events and their logical relationships are identified to construct the logical expression of the fault tree. The probability of the top event can be calculated quantitatively based on the probabilities of each risk factor. In this study, the fault tree is used to analyze cause-and-effect relationships between the identified factors, as well as to rank them by the probability of occurrence. The probability of the top event is evaluated using equations (3.20)–(3.22), which are derived from the principles of Boolean algebra:

$$P_{OR} = 1 - \prod_{i=1}^{n} (1 - P_i), \tag{3.20}$$

$$P_2 = \prod_{i=1}^{n} P_i, \tag{3.21}$$

$$P_{IE} = \prod_{j \in M} \left(1 - \prod_{BE_i \in Q_j}^{n} (1 - P_i) \right), \tag{3.22}$$

where P_i – the probability of occurrence of the basic event BE_i; Q_j – the set of basic events BE_i.

To assess the importance of each basic event, its contribution to the probability of the top event is determined. This information is highly valuable for decision-makers, as it allows identifying the most vulnerable points in the system. By doing so, decision support systems can effectively identify the factors most likely to lead to failures and require increased attention.

To identify and prioritize the most critical basic events leading to the top event, the Birnbaum importance measure is used. The Birnbaum importance measure is a key metric based on fault tree analysis and is used to assess the criticality of individual components or events in the system. It quantitatively evaluates the contribution of each basic event to the occurrence of the top event. Formally, the value of the Birnbaum importance measure for a specific basic event is defined as follows

$$IM_{BE_i}^{BIM} = P\left(IE \mid BR_i = 1\right) - P\left(IE \mid BR_i = 0\right), \tag{3.23}$$

where $IM_{BE_i}^{BIM}$ – the Birnbaum importance measure for the basic event BR_i.

Once the Birnbaum Importance Measure (BIM) values for all basic events are computed, they can be sorted according to their level of importance. A higher BIM value indicates a higher level of significance of the corresponding basic event with respect to the occurrence of the top event.

Action 6. Ranking of impact factors on the hierarchical system.

The use of fault tree analysis methods and interval-valued Pythagorean fuzzy sets ensures two types of weights: relative importance and corresponding ranking. To provide a balanced assessment that considers both the impact of the vulnerabilities and the likelihood of their occurrence, a corrective weight is introduced. This weight is used to reconcile both indicators, forming an updated ranking of the factors. Based on this new ranking, the most important factors are selected for further analysis. The updated ranking is calculated using the following formula

$$MR_i = w \cdot R_i^1 + (1 - w) \cdot R_i^2, \tag{3.24}$$

where MR_i – the combined ranking value for a factor i, R_i^1 – the ranking of factor i, obtained from the results of the fault tree analysis, R_i^2 – the rank of factor i, obtained from the results of the interval-valued

Pythagorean fuzzy sets, *w* – the corrective weight that defines the impact of each aspect. After this, the factors that cause failures can be re-sorted based on the combined ranking.

To determine the effectiveness of the proposed method, a simulation of its operation was conducted to solve the multi-criteria evaluation task of the state of the military grouping (forces) under the initial conditions specified in **Section 3.4.**

Separate parts of the computational experiment, using the proposed method, are presented in **Tables 3.3** and **3.4**. The overall computational experiment is detailed across more than 140 pages, with only a specific part of it presented in this section.

Table 3.3 Results of the calculation of membership functions for decisions based on rules

№	Results of the calculation of membership functions for decisions based on rules											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	0.0007	0.045	0.048	0.04	0.066	0.032	0.007	0.005	0.009	0.049	0.063	0.044
2	0.061	0.039	0.116	0	0.126	0.158	0.147	0.018	0.072	0.137	0.162	0.163
3	0.065	0.041	0.05	0.027	0.011	0.058	0.033	0.04	0.045	0.056	0.067	0.046
4	0.095	0.074	0.153	0.068	0.004	0.1	0.0018	0.169	0.0052	0.053	0.046	0.163
5	0.174	0.0147	0.083	0.083	0.076	0.002	0.102	0.083	0.162	0.116	0.09	0.105
6	0.028	0.057	0.019	0.036	0.047	0.038	0.025	0.028	0.0029	0.005	0.036	0.063
7	0.061	0.067	0.056	0.045	0.012	0.014	0.0007	0.012	0.022	0.056	0.069	0.00216
8	0.197	0.219	0.211	0.232	0.197	0.203	0.057	0.07	0.119	0.13	0.138	0.0054
9	0	0.122	0.124	0.157	0.243	0.003	0.262	0.208	0	0.165	0.084	0.151
10	0.146	0.079	0.142	0.076	0.005	0.121	0.107	0.121	0.114	0.091	0.049	0.139
11	0.165	0.139	0.065	0.044	0.07	0.1	0.083	0.163	0.061	0.165	0.133	0.086
12	0.026	0.039	0.001	0.006	0.043	0.021	0.036	0.013	0.014	0.034	0.02	0.03
13	0.035	0.006	0.037	0.04	0.021	0.038	0.004	0.0005	0.033	0.017	0.021	0.017
14	0.0054	0.003	0.033	0.021	0.007	0.028	0.029	0.0076	0.05	0.033	0.017	0.038
15	0.049	0.009	0.012	0.021	0.033	0.03	0.044	0.023	0.024	0.034	0.018	0.041
16	0.03	0.042	0.027	0.019	0.014	0.047	0.029	0.011	0.036	0.023	0.05	0.033
17	0.021	0.0005	0.031	0.028	0.032	0.047	0.031	0.02	0.024	0.012	0.02	0.032
18	0.03	0.008	0.016	0.044	0.02	0.036	0.016	0.048	0.05	0.014	0.035	0.0086
19	0.026	0.039	0.038	0.014	0.003	0.002	0.031	0.011	0.031	0.0076	0.034	0.013
20	0.007	0.046	0.049	0.033	0.015	0.007	0.049	0.023	0.05	0.016	0.03	0.034
21	0.042	0.026	0.026	0.025	0.037	0.029	0.027	0.021	0.015	0.01	0.041	0.00758

Continuation of Table 3.3

	1	2	3	4	5	6	7	8	9	10	11	12
22	0.126	0.027	0.017	0.315	0.033	0.096	0.206	0.305	0.093	0.146	0.116	0.00332
23	0.391	0.462	0.616	0.443	0.077	0.231	0.0064	0.077	0.616	0.109	0.237	0.61
24	0.132	0.005	0.04	0.002	0.035	0.139	0.063	0.0088	0.112	0.118	0.109	0.037
25	0.14	0.125	0.044	0.139	0.13	0.074	0.107	0.125	0.1	0.054	0.021	0.158
26	0.041	0.047	0.02	0.026	0.008	0.016	0.025	0.019	0.043	0.031	0.04	0.049
27	0.022	0.014	0.041	0.037	0.034	0.046	0.013	0.027	0.022	0.011	0.042	0.012
28	0.038	0.008	0.015	0.011	0.018	0	0.017	0.033	0.018	0.042	0.043	0.023
29	0.037	0	0.039	0.015	0.035	0.004	0.021	0.017	0.039	0.031	0.004	0.05
30	0.007	0.028	0.011	0.031	0.012	0.048	0.021	0.026	0.032	0.036	0.033	0.026
31	0.032	0.011	0.007	0.018	0.033	0.036	0.04	0.011	0.038	0.024	0.018	0.045
32	0.041	0.02	0.05	0.027	0.008	0.017	0.05	0.024	0.031	0.045	0.034	0.022
33	0.022	0.019	0.039	0.049	0.043	0.000	0.045	0.029	0.0025	0.016	0.013	0.037
34	0.042	0.048	0.011	0.02	0.013	0.042	0.006	0.0035	0.014	0.0056	0.049	0.049
35	0.05	0.032	0.032	0.037	0.027	0.014	0.005	0.046	0.038	0.02	0.037	0.039
36	0.081	0.044	0.049	0.102	0.016	0.146	0.053	0.114	0.133	0.054	0.054	0.086
37	0.139	0.153	0.025	0.172	0.014	0.142	0.025	0.114	0.063	0.04	0.091	0.135
38	0.019	0.044	0.012	0.004	0.03	0.047	0.008	0.024	0.05	0.033	0.008	0.0015
39	0.023	0.034	0.041	0.003	0.015	0.015	0.05	0.048	0.018	0.036	0.035	0.027
40	0.034	0.063	0.056	0.023	0.085	0.045	0.025	0.0073	0.012	0.113	0.078	0.036
41	0.045	0.016	0.023	0.027	0.032	0.006	0.027	0.011	0.036	0.045	0.038	0.041
42	0.018	0.013	0.019	0.038	0.05	0.021	0.023	0.03	0.028	0.024	0.015	0.045
43	0.0005	0.031	0.033	0.028	0.047	0.023	0.0005	0.035	0.0066	0.034	0.044	0.031
Defense	0.174	0.147	0.153	0.083	0.126	0.158	0.147	0.169	0.162	0.137	0.162	0.163
Counteroffensive	0.391	0.462	0.616	0.443	0.243	0.231	0.262	0.305	0.616	0.165	0.237	0.61
Stabilization actions	0.139	0.153	0.056	0.172	0.14	0.142	0.05	0.114	0.063	0.113	0.091	0.135
Error	0.42	0.334	0.174	0.347	0.609	0.636	0.569	0.525	0.178	0.729	0.617	0.197

Table 3.4 Comparative results of the state evaluation process for the troop (force) grouping

	Using the method	Without using the method
Operational efficiency of the evaluation process		
Best case, sec.	49 – 303	56 – 507.1
Worst case, sec.	255.1 – 2501.5	382.8 – 3977
Reliability of the obtained decisions		
Best case, sec.	0.89 – 1.0	0.64 – 0.85
Worst case, sec.	0.77 – 1.0	0.617 – 0.75

From the analysis of **Table 3.4**, it can be concluded that the proposed method provides an average increase in accuracy and operational efficiency by 35%, while ensuring high convergence of the obtained results at a level of 93.17%.

CONCLUSIONS

The algorithm for implementing the methodology has been defined, thanks to additional and improved procedures, which allow:

– verification of the topology and parameters of decision support systems, taking into account the degree of uncertainty in the initial data regarding the known information, through the use of the enhanced penguin swarm algorithm. This allows reducing the time for initial setup during the first configuration of the evaluation methodology;

– preliminary selection of individuals for configuring the evolving artificial neural network, carried out using the enhanced genetic algorithm, which reduces solution search time and increases the reliability of the obtained solutions;

– exploration of solution spaces for parameter evaluation problems in decision support systems, described by atypical functions, using the enhanced penguin swarm algorithm;

– configuration of the weights of the evolving artificial neural network, leading to increased accuracy in evaluating decision support system parameters;

– use of additional mechanisms to adjust the parameters of the evolving artificial neural network through the procedure of modifying the membership function;

– increased reliability in evaluating decision support system parameters by parallel evaluation using multiple evaluation methods;

– use of hybrid parameter evaluation for decision support systems, enabling proper operation in the absence of conditions for stationarity, homogeneity, normality, and independence;

– simultaneous search for solutions in different directions;

– calculation of the required number of computational resources needed when existing resources are insufficient for calculations.

An example of using the proposed methodology for evaluating decision support system parameters showed an increase in the reliability of parameter evaluation by 17–21% through the use of additional procedures, while maintaining the required operational speed.

The algorithm for implementing the method has been determined, thanks to additional and improved procedures, which allow:

– verification of the input data and clarification of relationships between elements in the hierarchical system using the enhanced penguin swarm algorithm. This minimizes the error of entering incorrect data for the operational grouping of troops (forces);

– description of the external and internal factors affecting the hierarchical system, which is subject to multi-criteria evaluation using fuzzy cognitive models;

– adaptation to the hierarchical system type through multi-level adaptation of the indicator system and evaluation criteria;

– reduction of uncertainty using interval Pythagorean fuzzy sets, which improves the reliability of multi-criteria evaluation of the state of hierarchical systems;

– identification of the most vulnerable elements of the hierarchical system using a fault tree;

– adaptation of the membership function type based on the available computational resources, ensuring adaptation to the available computational resources.

An example of using the proposed method for multi-criteria evaluation of the operational grouping of troops (forces) was provided, showing that the proposed method ensures an average increase in accuracy and operational speed by 35%, while ensuring high convergence of the results at a level of 93.17%.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

USE OF ARTIFICIAL INTELLIGENCE

The authors confirm that they did not use artificial intelligence technologies in creating the submitted work.

REFERENCES

1. Dudnyk, V., Sinenko, Y., Matsyk, M., Demchenko, Y., Zhyvotovskyi, R., Repilo, I. et al. (2020). Development of a method for training artificial neural networks for intelligent decision support systems. Eastern-European Journal of Enterprise Technologies, 3 (2 (105)), 37–47. https://doi.org/10.15587/1729-4061.2020.203301

2. Sova, O., Shyshatskyi, A., Salnikova, O., Zhuk, O., Trotsko, O., Hrokholskyi, Y. (2021). Development of a method for assessment and forecasting of the radio electronic environment. EUREKA: Physics and Engineering, 4, 30–40. https://doi.org/10.21303/2461-4262.2021.001940
3. Pievtsov, H., Turinskyi, O., Zhyvotovskyi, R., Sova, O., Zvieriev, O., Lanetskii, B., Shyshatskyi, A. (2020). Development of an advanced method of finding solutions for neuro-fuzzy expert systems of analysis of the radioelectronic situation. EUREKA: Physics and Engineering, 4, 78–89. https://doi.org/10.21303/2461-4262.2020.001353
4. Zuiev, P., Zhyvotovskyi, R., Zvieriev, O., Hatsenko, S., Kuprii, V., Nakonechnyi, O. et al. (2020). Development of complex methodology of processing heterogeneous data in intelligent decision support systems. Eastern-European Journal of Enterprise Technologies, 4 (9 (106)), 14–23. https://doi.org/10.15587/1729-4061.2020.208554
5. Kuchuk, N., Mohammed, A. S., Shyshatskyi, A., Nalapko, O. (2019). The Method of Improving the Efficiency of Routes Selection in Networks of Connection with the Possibility of Self-Organization. International Journal of Advanced Trends in Computer Science and Engineering, 8 (1.2), 1–6.
6. Shyshatskyi, A., Zvieriev, O., Salnikova, O., Demchenko, Ye., Trotsko, O., Neroznak, Ye. (2020). Complex Methods of Processing Different Data in Intellectual Systems for Decision Support System. International Journal of Advanced Trends in Computer Science and Engineering, 9 (4), 5583–5590. https://doi.org/10.30534/ijatcse/2020/206942020
7. Pozna, C., Precup, R.-E., Horvath, E., Petriu, E. M. (2022). Hybrid Particle Filter–Particle Swarm Optimization Algorithm and Application to Fuzzy Controlled Servo Systems. IEEE Transactions on Fuzzy Systems, 30 (10), 4286–4297. https://doi.org/10.1109/tfuzz.2022.3146986
8. Yang, X.-S., Deb, S. (2013). Cuckoo search: recent advances and applications. Neural Computing and Applications, 24 (1), 169–174. https://doi.org/10.1007/s00521-013-1367-1
9. Mirjalili, S. (2015). The Ant Lion Optimizer. Advances in Engineering Software, 83, 80–98. https://doi.org/10.1016/j.advengsoft.2015.01.010
10. Yu, J. J. Q., Li, V. O. K. (2015). A social spider algorithm for global optimization. Applied Soft Computing, 30, 614–627. https://doi.org/10.1016/j.asoc.2015.02.014
11. Mirjalili, S., Mirjalili, S. M., Lewis, A. (2014). Grey Wolf Optimizer. Advances in Engineering Software, 69, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
12. Koval, V., Nechyporuk, O., Shyshatskyi, A., Nalapko, O., Shknai, O., Zhyvylo, Y. et al. (2023). Improvement of the optimization method based on the cat pack algorithm. Eastern-European Journal of Enterprise Technologies, 1 (9 (121)), 41–48. https://doi.org/10.15587/1729-4061.2023.273786
13. Gupta, E., Saxena, A. (2015). Robust generation control strategy based on Grey Wolf Optimizer. Journal of Electrical Systems, 11, 174–188.
14. Chaman-Motlagh, A. (2015). Superdefect Photonic Crystal Filter Optimization Using Grey Wolf Optimizer. IEEE Photonics Technology Letters, 27 (22), 2355–2358. https://doi.org/10.1109/lpt.2015.2464332
15. Nuaekaew, K., Artrit, P., Pholdee, N., Bureerat, S. (2017). Optimal reactive power dispatch problem using a two-archive multi-objective grey wolf optimizer. Expert Systems with Applications, 87, 79–89. https://doi.org/10.1016/j.eswa.2017.06.009

16. Koval, M., Sova, O., Orlov, O., Shyshatskyi, A., Artabaiev, Y., Shknai, O. et al. (2022). Improvement of complex resource management of special-purpose communication systems. Eastern-European Journal of Enterprise Technologies, 5 (9 (119)), 34–44. https://doi.org/10.15587/1729-4061.2022.266009
17. Ali, M., El-Hameed, M. A., Farahat, M. A. (2017). Effective parameters' identification for polymer electrolyte membrane fuel cell models using grey wolf optimizer. Renewable Energy, 111, 455–462. https://doi.org/10.1016/j.renene.2017.04.036
18. Zhang, S., Zhou, Y. (2017). Template matching using grey wolf optimizer with lateral inhibition. Optik, 130, 1229–1243. https://doi.org/10.1016/j.ijleo.2016.11.173
19. Khouni, S. E., Menacer, T. (2023). Nizar optimization algorithm: a novel metaheuristic algorithm for global optimization and engineering applications. The Journal of Supercomputing, 80 (3), 3229–3281. https://doi.org/10.1007/s11227-023-05579-4
20. Saremi, S., Mirjalili, S., Lewis, A. (2017). Grasshopper Optimisation Algorithm: Theory and application. Advances in Engineering Software, 105, 30–47. https://doi.org/10.1016/j.advengsoft.2017.01.004
21. Braik, M. S. (2021). Chameleon Swarm Algorithm: A bio-inspired optimizer for solving engineering design problems. Expert Systems with Applications, 174, 114685. https://doi.org/10.1016/j.eswa.2021.114685
22. Thamer, K. A., Sova, O., Shaposhnikova, O., Yashchenok, V., Stanovska, I., Shostak, S. et al. (2024). Development of a solution search method using a combined bio-inspired algorithm. Eastern-European Journal of Enterprise Technologies, 1 (4 (127)), 6–13. https://doi.org/10.15587/1729-4061.2024.298205
23. Yapici, H., Cetinkaya, N. (2019). A new meta-heuristic optimizer: Pathfinder algorithm. Applied Soft Computing, 78, 545–568. https://doi.org/10.1016/j.asoc.2019.03.012
24. Duan, H., Qiao, P. (2014). Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning. International Journal of Intelligent Computing and Cybernetics, 7 (1), 24–37. https://doi.org/10.1108/ijicc-02-2014-0005
25. Shyshatskyi, A., Romanov, O., Shknai, O., Babenko, V., Koshlan, O., Pluhina, T. et al. (2023). Development of a solution search method using the improved emperor penguin algorithm. Eastern-European Journal of Enterprise Technologies, 6 (4 (126)), 6–13. https://doi.org/10.15587/1729-4061.2023.291008
26. Yang, X. S (2012). Flower pollination algorithm for global optimization. Unconventional computing and natural computation. Berlin, Heidelberg: Springer, 240–249. https://doi.org/10.1007/978-3-642-32894-7_27
27. Gomes, G. F., da Cunha, S. S., Ancelotti, A. C. (2018). A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates. Engineering with Computers, 35 (2), 619–626. https://doi.org/10.1007/s00366-018-0620-8
28. Mehrabian, A. R., Lucas, C. (2006). A novel numerical optimization algorithm inspired from weed colonization. Ecological Informatics, 1 (4), 355–366. https://doi.org/10.1016/j.ecoinf.2006.07.003
29. Qi, X., Zhu, Y., Chen, H., Zhang, D., Niu, B. (2013). An idea based on plant root growth for numerical optimization. Intelligent Computing Theories and Technology. Berlin, Heidelberg: Springer, 571–578. https://doi.org/10.1007/978-3-642-39482-9_66
30. Bezuhlyi, V., Oliynyk, V., Romanenko, I., Zhuk, O., Kuzavkov, V., Borysov, O., Korobchenko, S., Ostapchuk, E., Davydenko, T., Shyshatskyi, A. (2021). Development of object state estimation method in intelligent

decision support systems. Eastern-European Journal of Enterprise Technologies, 5 (3 (113)), 54–64. https://doi.org/10.15587/1729-4061.2021.239854

31. Mahdi, Q. A., Shyshatskyi, A., Prokopenko, Y., Ivakhnenko, T., Kupriyenko, D., Golian, V. et al. (2021). Development of estimation and forecasting method in intelligent decision support systems. Eastern-European Journal of Enterprise Technologies, 3 (9 (111)), 51–62. https://doi.org/10.15587/1729-4061.2021.232718
32. Sova, O., Radzivilov, H., Shyshatskyi, A., Shevchenko, D., Molodetskyi, B., Stryhun, V. et al. (2022). Development of the method of increasing the efficiency of information transfer in the special purpose networks. Eastern-European Journal of Enterprise Technologies, 3 (4 (117)), 6–14. https://doi.org/10.15587/1729-4061.2022.259727
33. Zhang, H., Zhu, Y., Chen, H. (2013). Root growth model: a novel approach to numerical function optimization and simulation of plant root system. Soft Computing, 18 (3), 521–537. https://doi.org/10.1007/s00500-013-1073-z
34. Labbi, Y., Attous, D. B., Gabbar, H. A., Mahdad, B., Zidan, A. (2016). A new rooted tree optimization algorithm for economic dispatch with valve-point effect. International Journal of Electrical Power & Energy Systems, 79, 298–311. https://doi.org/10.1016/j.ijepes.2016.01.028
35. Murase, H. (2000). Finite element inverse analysis using a photosynthetic algorithm. Computers and Electronics in Agriculture, 29 (1-2), 115–123. https://doi.org/10.1016/s0168-1699(00)00139-3
36. Zhao, S., Zhang, T., Ma, S., Chen, M. (2022). Dandelion Optimizer: A nature-inspired metaheuristic algorithm for engineering applications. Engineering Applications of Artificial Intelligence, 114, 105075. https://doi.org/10.1016/j.engappai.2022.105075
37. Paliwal, N., Srivastava, L., Pandit, M. (2020). Application of grey wolf optimization algorithm for load frequency control in multi-source single area power system. Evolutionary Intelligence, 15 (1), 563–584. https://doi.org/10.1007/s12065-020-00530-5
38. Dorigo, M., Blum, C. (2005). Ant colony optimization theory: A survey. Theoretical Computer Science, 344 (2-3), 243–278. https://doi.org/10.1016/j.tcs.2005.05.020
39. Poli, R., Kennedy, J., Blackwell, T. (2007). Particle swarm optimization. Swarm Intelligence, 1 (1), 33–57. https://doi.org/10.1007/s11721-007-0002-0
40. Bansal, J. C., Sharma, H., Jadon, S. S., Clerc, M. (2014). Spider Monkey Optimization algorithm for numerical optimization. Memetic Computing, 6 (1), 31–47. https://doi.org/10.1007/s12293-013-0128-0
41. Yeromina, N., Kurban, V., Mykus, S., Peredrii, O., Voloshchenko, O., Kosenko, V. et al. (2021). The Creation of the Database for Mobile Robots Navigation under the Conditions of Flexible Change of Flight Assignment. International Journal of Emerging Technology and Advanced Engineering, 11 (5), 37–44. https://doi.org/10.46338/ijetae0521_05
42. Maccarone, A. D., Brzorad, J. N., Stone H. M. (2008). Characteristics And Energetics Of Great Egret And Snowy Egret Foraging Flights. Waterbirds, 31 (4), 541–549. https://doi.org/10.1675/1524-4695-31.4.541
43. Ramaji, I. J., Memari, A. M. (2018). Interpretation of structural analytical models from the coordination view in building information models. Automation in Construction, 90, 117–133. https://doi.org/10.1016/j.autcon.2018.02.025

44. Pérez-González, C. J., Colebrook, M., Roda-García, J. L., Rosa-Remedios, C. B. (2019). Developing a data analytics platform to support decision making in emergency and security management. Expert Systems with Applications, 120, 167–184. https://doi.org/10.1016/j.eswa.2018.11.023
45. Chen, H. (2018). Evaluation of Personalized Service Level for Library Information Management Based on Fuzzy Analytic Hierarchy Process. Procedia Computer Science, 131, 952–958. https://doi.org/10.1016/j.procs.2018.04.233
46. Chan, H. K., Sun, X., Chung, S.-H. (2019). When should fuzzy analytic hierarchy process be used instead of analytic hierarchy process? Decision Support Systems, 125, 113114. https://doi.org/10.1016/j.dss.2019.113114
47. Osman, A. M. S. (2019). A novel big data analytics framework for smart cities. Future Generation Computer Systems, 91, 620–633. https://doi.org/10.1016/j.future.2018.06.046
48. Nechyporuk, O., Sova, O., Shyshatskyi, A., Kravchenko, S., Nalapko, O., Shknai, O. et al. (2023). Development of a method of complex analysis and multidimensional forecasting of the state of intelligence objects. Eastern-European Journal of Enterprise Technologies, 2 (4 (122)), 31–41. https://doi.org/10.15587/1729-4061.2023.276168
49. Merrikh-Bayat, F. (2015). The runner-root algorithm: A metaheuristic for solving unimodal and multimodal optimization problems inspired by runners and roots of plants in nature. Applied Soft Computing, 33, 292–303. https://doi.org/10.1016/j.asoc.2015.04.048
50. Poliarush, O., Krepych, S., Spivak, I. (2023). hybrid approach for data filtering and machine learning inside content management system. Advanced Information Systems, 7 (4), 70–74. https://doi.org/10.20998/2522-9052.2023.4.09
51. Balochian, S., Baloochian, H. (2019). Social mimic optimization algorithm and engineering applications. Expert Systems with Applications, 134, 178–191. https://doi.org/10.1016/j.eswa.2019.05.035
52. Lenord Melvix, J. S. M. (2014). Greedy Politics Optimization: Metaheuristic inspired by political strategies adopted during state assembly elections. 2014 IEEE International Advance Computing Conference (IACC), 1157–1162. https://doi.org/10.1109/iadcc.2014.6779490
53. Moosavian, N., Roodsari, B. K. (2014). Soccer League Competition Algorithm, a New Method for Solving Systems of Nonlinear Equations. International Journal of Intelligence Science, 4 (1), 7–16. https://doi.org/10.4236/ijis.2014.41002
54. Hayyolalam, V., Pourhaji Kazem, A. A. (2020). Black Widow Optimization Algorithm: A novel meta-heuristic approach for solving engineering optimization problems. Engineering Applications of Artificial Intelligence, 87, 103249. https://doi.org/10.1016/j.engappai.2019.103249
55. Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-qaness, M. A. A., Gandomi, A. H. (2021). Aquila Optimizer: A novel meta-heuristic optimization algorithm. Computers & Industrial Engineering, 157, 107250. https://doi.org/10.1016/j.cie.2021.107250
56. Hodlevskyi, M., Burlakov, G. (2023). Information technology of quality improvement planning of process subsets of the spice model. Advanced Information Systems, 7 (4), 52–59. https://doi.org/10.20998/2522-9052.2023.4.06
57. Askari, Q., Younas, I., Saeed, M. (2020). Political Optimizer: A novel socio-inspired meta-heuristic for global optimization. Knowledge-Based Systems, 195, 105709. https://doi.org/10.1016/j.knosys.2020.105709

58. Mohamed, A. W., Hadi, A. A., Mohamed, A. K. (2019). Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm. International Journal of Machine Learning and Cybernetics, 11 (7), 1501–1529. https://doi.org/10.1007/s13042-019-01053-x
59. Gödri, I., Kardos, C., Pfeiffer, A., Váncza, J. (2019). Data analytics-based decision support workflow for high-mix low-volume production systems. CIRP Annals, 68 (1), 471–474. https://doi.org/10.1016/j.cirp.2019.04.001
60. Harding, J. L. (2013). Data quality in the integration and analysis of data from multiple sources: some research challenges. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XL-2/W1, 59–63. https://doi.org/10.5194/isprsarchives-xl-2-w1-59-2013
61. Orouskhani, M., Orouskhani, Y., Mansouri, M., Teshnehlab, M. (2013). A Novel Cat Swarm Optimization Algorithm for Unconstrained Optimization Problems. International Journal of Information Technology and Computer Science, 5 (11), 32–41. https://doi.org/10.5815/ijitcs.2013.11.04
62. Karaboga, D., Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization, 39 (3), 459–471. https://doi.org/10.1007/s10898-007-9149-x
63. Fister, I., Fister, I., Yang, X.-S., Brest, J. (2013). A comprehensive review of firefly algorithms. Swarm and Evolutionary Computation, 13, 34–46. https://doi.org/10.1016/j.swevo.2013.06.001
64. Sova, O., Radzivilov, H., Shyshatskyi, A., Shvets, P., Tkachenko, V., Nevhad, S. et al. (2022). Development of a method to improve the reliability of assessing the condition of the monitoring object in special-purpose information systems. Eastern-European Journal of Enterprise Technologies, 2 (3 (116)), 6–14. https://doi.org/10.15587/1729-4061.2022.254122
65. Khudov, H., Khizhnyak, I., Glukhov, S., Shamrai, N., Pavlii, V. (2024). The method for objects detection on satellite imagery based on the firefly algorithm. Advanced Information Systems, 8 (1), 5–11. https://doi.org/10.20998/2522-9052.2024.1.01
66. Owaid, S. R., Zhuravskyi, Y., Lytvynenko, O., Veretnov, A., Sokolovskyi, D., Plekhova, G. et al. (2024). Development of a method of increasing the efficiency of decision-making in organizational and technical systems. Eastern-European Journal of Enterprise Technologies, 1 (4 (127)), 14–22. https://doi.org/10.15587/1729-4061.2024.298568
67. Tyurin, V., Bieliakov, R., Odarushchenko, E., Yashchenok, V., Shaposhnikova, O., Lyashenko, A. et al. (2023). Development of a solution search method using an improved locust swarm algorithm. Eastern-European Journal of Enterprise Technologies, 5 (4 (125)), 25–33. https://doi.org/10.15587/1729-4061.2023.287316
68. Yakymiak, S., Vdovytskyi, Y., Artabaiev, Y., Degtyareva, L., Vakulenko, Y., Nevhad, S. et al. (2023). Development of the solution search method using the population algorithm of global search optimization. Eastern-European Journal of Enterprise Technologies, 3 (4 (123)), 39–46. https://doi.org/10.15587/1729-4061.2023.281007
69. Mohammed, B. A., Zhuk, O., Vozniak, R., Borysov, I., Petrozhalko, V., Davydov, I. et al. (2023). Improvement of the solution search method based on the cuckoo algorithm. Eastern-European Journal of Enterprise Technologies, 2 (4 (122)), 23–30. https://doi.org/10.15587/1729-4061.2023.277608

70. Raskin, L., Sukhomlyn, L., Sokolov, D., Vlasenko, V. (2023). Multi-criteria evaluation of the multifactor stochastic systems effectiveness. Advanced Information Systems, 7 (2), 63–67. https://doi.org/10.20998/2522-9052.2023.2.09
71. Arora, S., Singh, S. (2018). Butterfly optimization algorithm: a novel approach for global optimization. Soft Computing, 23 (3), 715–734. https://doi.org/10.1007/s00500-018-3102-4