DOI: 10.15587/978-617-8360-13-9.CH3

Oleh Shknai, Oleg Sova, Olena Nechyporuk,
Oleksii Nalapko, Oleksiy Buyalo, Anna Lyashenko

CHAPTER 3

A SET OF METHODS FOR ENHANCING THE EFFICIENCY OF INFORMATION PROCESSING IN INTELLIGENT DECISION SUPPORT SYSTEMS

ABSTRACT

This section of the study proposes a set of methods to enhance the efficiency of information processing in intelligent decision support systems.

The authors suggest the following methods:

– a method for managing information flows in intelligent decision support systems using a population-based algorithm;

– a method for evaluating the timeliness of processing diverse data types in decision support systems;

– a method for assessment and forecasting in intelligent decision support systems.

The novelty of the proposed methods lies in:

– determining the initial population of agents and their starting positions in the search space by considering the degree of uncertainty in the initial data about information flows within intelligent decision support systems;

– accounting for the initial velocity of each agent, which enables prioritization of search tasks within the respective search space;

– universality of agent feeding search strategies, allowing the classification of conditions and factors that influence the management of information flows in intelligent decision support systems;

– ability to explore solution spaces described by atypical functions through the application of agent movement technique selection procedures;

– capability to search for solutions simultaneously in multiple directions;

– potential for deep learning of agents' knowledge bases;

– ability to calculate the required amount of computational resources to be engaged in cases where existing resources are insufficient for necessary computations;

– consideration of the type of uncertainty in the data circulating within decision support systems;

– implementation of adaptive strategies for solution space searches by the population agents;

– prioritization of search tasks by population agents;

– initial placement of population members considering the type of uncertainty;

– application as a universal tool for analyzing the timeliness of processing diverse data types in decision support systems;

– verification of the adequacy of the obtained results;

– avoidance of the local extremum problem;

– use of a new type of fuzzy cognitive temporal models focused on multidimensional analysis and forecasting of object states under conditions of uncertainty;

– ability to combine elements of artificial neural networks;

– capability to train individual elements of artificial neural networks;

– data computation within a single epoch without the need to store previous calculations;

– prevention of error accumulation during the training of artificial neural networks as a result of processing incoming information.

KEYWORDS

Artificial intelligence, processing of diverse data, decision support systems, reliability, timeliness.

3.1 DEVELOPMENT OF A METHOD FOR MANAGING INFORMATION FLOWS IN INTELLIGENT DECISION SUPPORT SYSTEMS USING A POPULATION-BASED ALGORITHM

The method for managing information flows in intelligent decision support systems (IDSS) using a population-based algorithm consists of the following sequence of actions:

Action 1. Input of initial data.

At this stage, the initial data regarding IDSS information sources, available communication channels for information flows, and other relevant parameters are entered [1–18].

Action 2. Initialization and formation of the ABC group.

In this step, initial random sets of solutions are generated to represent the ABC groups. These groups are marked with the "scent" attributes of the individuals' feet within the ABC group, which are considered decision variables from a set of potential solutions. The mathematical representation of a randomly selected ABC group within a defined territory is expressed by the following equation:

$$P_{i,j} = P_{i,j}^{\min} + \left(\lambda\left(P_{i,j}^{\max} - P_{i,j}^{\min}\right)\right)\gamma, \tag{3.1}$$

where λ – a random number in the range from 0 to 1; $P_{i,j}$ – i-th designation of the paw in the j-th group of ABC. The process involves arranging the set of ABCs in ascending order of $f(P_i)$, selecting the best $\left(P_{i,j}^{best}\right)$ and the worst solutions $\left(P_i^{worst}\right)$; γ – represents the degree of uncertainty in the data about information flows in the DSS. At this stage, the objective function $f(P)$, the population size (m) of the ABC swarm, the number of variables (n), constraints on variable values (LB, UB) and the termination criterion of the algorithm ($FE_{\max}$'). The group of brown bears is considered part of the ABC population ($i = 1,2,\ldots,m$), and the ABC levels in the group are treated as decision variables ($j = 1,2,\ldots,n$).

CHAPTER 3

Action 3. Assigning numbers to the ABCs in the population, i, $i \in [0, S]$.

At this stage, each ABC in the population is assigned a serial number [19–36].

Action 4. Determining the initial velocity of the ABCs in the population.

The initial velocity v_0 of each ABC in the population is determined by the following expression:

$$v_i = \left(v_1, v_2 \ldots v_S\right),\ v_i = v_0. \tag{3.2}$$

In the proposed approach, the position of the ABCs in the problem-solving space is updated based on modeling exploration and exploitation strategies.

Action 5. Preliminary evaluation of the ABC search area.

In this procedure, the search area is defined in natural language as the habitat of the ABCs. Given the diversity of ABCs' food sources, the quality of food is sorted accordingly.

Action 6. Classification of food sources for ABCs.

The location of the best food source (i.e., minimum fitness) is considered (FS_{ht}) which represents plant-based food: berries, acorns, nuts, roots, and grass tubers found nearby and requiring the least energy to locate and obtain. Delicacies such as honey are denoted as FS_{at}.

Other non-priority food sources (necessary for the survival of individuals) are denoted as FS_{nt}:

$$FS_{ht} = FS(\text{sorte\_index}(1)), \tag{3.3}$$

$$FS_{at}(1{:}4) = FS(\text{sorte\_index}(1{:}4)), \tag{3.4}$$

$$FS_{nt}(1{:}NP-4) = FS(\text{sorte\_index}(4{:}NP)). \tag{3.5}$$

Action 7. Selection of ABC walking techniques.

Action 7.1. Foot rotation during ABC walking.

This action involves a specific walking behavior where ABCs rotate their feet to avoid previous depressions on the ground and carefully step towards the required location. This unique walking behavior is most observed in male ABCs. Mathematically, this behavior can be modeled as:

$$P_{i,j,k}^{new} = P_{i,j,k}^{old} - \left(\theta_k \alpha_{i,j,k} P_{i,j,k}^{old}\right), \tag{3.6}$$

where $P_{i,j,k}^{new}$ – the updated paw scent mark at the k-th iteration of the i-th group, created by the j-th paw mark; θ_k – the repetition coefficient, taking values from 0 to 1; $\alpha_{i,j,k}$ – a random number in the range from 0 to 1.

Action 7.2. Careful step of ABCs.

The characteristic of the careful step involves repeating paw prints by verifying previously marked paw tracks. This behavior helps effectively warn other group members. Equation (3.7) represents the mathematical formulation of the careful step technique observed in ABCs:

$$P_{i,j,k}^{new} = P_{i,j,k}^{old} + F_k\left(P_{j,k}^{best} - L_k P_{j,k}^{worst}\right), \tag{3.7}$$

where F_k – the step coefficient; $P_{j,k}^{best}, P_{j,k}^{worst}$ – j-th best and worst evaluations of the ABC paw at the k-th iteration; L_k – the step length of the ABC at a given iteration.

Action 7.3. Paw twisting by ABCs.

The third unique walking behavior observed in brown bears is paw twisting. Male brown bears typically twist their feet into pre-formed paw tracks, making them deeper and more prominent for easier identification. The choice of the paw track type is based on the best and worst scent tracks of the ABC paw determined in the previous iteration. The mathematical representation of the paw twisting behavior is described by equation (3.8):

$$P_{i,j,k}^{new} = P_{i,j,k}^{old} + \omega_{i,k}\left(P_{j,k}^{best} - P_{i,j,k}^{old}\right) - \omega_{i,k}\left(P_{j,k}^{worst} - P_{i,j,k}^{old}\right), \tag{3.8}$$

where $\omega_{i,k}$ – angular velocity of legs at the i-th paw mark and the k-th iteration.

Action 8. Placement of the ABC set.

At this stage, the ABCs are arranged in ascending order of the objective function values $f(P_i)$, and the best and worst solutions are selected $\left(P_i^{best}\right)$ and the best and worst solutions are selected $\left(P_i^{worst}\right)$.

Action 9. Creation of a new ABC population:

$$P_{i,k}^{new} = P_{i,k}^{old} - \left(\theta_k \alpha_{i,k} P_{i,k}^{old}\right), \tag{3.9}$$

where $F_k = \beta_{1,k} \cdot \theta_k, L_k = 1 + \beta_{2,k}$ and $\beta_{1,k} \cdot \beta_{2,k}$ – random numbers in the range [0, 1].

Action 10. Placement of the ABC set.

At this stage, the ABCs are again arranged in ascending order of the objective function values $f(P_i)$ and the best $\left(P_i^{best}\right)$ and worst solutions are selected $\left(P_i^{worst}\right)$.

Action 11. ABC sniffing.

The essence of ABC behavior involves sniffing each other to follow the scent trails of group members in the correct direction. In addition, ABCs use sniffing to establish their territory and avoid being misled by the scent trails of other ABCs. The mathematical model for sniffing behavior is given by formula (3.10):

$$P_{m,j,k}^{new} = \begin{cases} P_{m,j,k}^{old} + \lambda_{i,k}\left(P_{m,j,k}^{old} - P_{n,j,k}^{old}\right), \ if \ f\left(P_{m,k}^{old}\right) \langle f\left(P_{n,k}^{old}\right), \\ P_{m,j,k}^{old} + \lambda_{i,k}\left(P_{n,j,k}^{old} - P_{m,j,k}^{old}\right), \ if \ f\left(P_{n,k}^{old}\right) \langle f\left(P_{m,k}^{old}\right), \end{cases} \tag{3.10}$$

where $\lambda_{i,k}$ – a uniformly distributed random number in the range 0 to 1; $P_{m,j,k}^{new}$ – updated location of the scent on the paw with $m \neq n$; $P_{m,k}^{old}$ and $P_{n,k}^{old}$ – the values of the fitness function at the k-th iteration of m and n groups, respectively. The update process described for all stages is applied to each ABC group until the necessary termination criterion is met.

CHAPTER 3

Action 12. Termination criteria check.

The algorithm terminates if the maximum number of iterations is reached. Otherwise, the behavior for generating new locations and verifying conditions is repeated.

Action 13. Training the ABC knowledge base.

In this study, the training of each ABC's knowledge base uses a method based on artificial neural networks that evolve, as proposed in [2]. This method modifies the movement patterns of each ABC for more accurate analysis results in subsequent stages.

Action 14. Determination of required computational resources for the intelligent decision support system.

To avoid computational looping in *Actions 1–10* of this method and to improve computational efficiency, the system load is additionally assessed. If the defined threshold of computational complexity is exceeded, the number of software and hardware resources needed is determined using the method proposed in [31].

End of the algorithm.

The proposed method for managing information flows in the DSS uses a population-based algorithm. To evaluate its efficiency, the method was simulated to solve the problem of information flow management in the DSS [37–58].

The efficiency of the information flow management method in the DSS using the population-based algorithm is compared using the functions presented in **Table 3.1**.

Table 3.1 Assessment of the effectiveness of the proposed management method by the criterion of information processing speed

Function Name	Metric	Particle Swarm Algorithm	Ant Colony Algorithm	Black Widow Algorithm	Grey Wolf Swarm Algorithm	Canonical Brown Bear Algorithm	Improved Brown Bear Algorithm
1	2	3	4	5	6	7	8
U22-1	Average value	300.000	300.000	300.000	300.000	300.000	300.000
	Standard value	2.17547E-07	1.94448E-07	1.73866E-07	1.73121E-07	1.51021E-07	1.72168E-07
B22-2	Average value	400	400.265772	400.7973158	400.265772	400.3986579	400.5315439
	Standard value	4.9898E-08	1.011427534	1.621892282	1.011427535	1.216419212	1.378343398
B22-3	Average value	600.0071815	600.0644622	600.0240021	600.012832	600.031303	600.0449987
	Standard value	0.021632777	0.184980091	0.115606243	0.053463097	0.147011513	0.101164243

Continuation of Table 3.1

1	2	3	4	5	6	7	8
B22-4	Average value	826.5653461	827.3281442	823.8789639	826.3000191	826.2668486	825.7693662
	Standard value	9.13817552	8.364210734	11.30806963	8.186625055	9.136107323	10.05991317
B22-5	Average value	900.743876	900.9504411	900.9726169	900.8007883	900.5452042	901.2016312
	Standard value	0.781626306	1.424558753	1.275779755	0.903385622	0.635781924	1.598982565
B22-6	Average value	1888.524629	1874.869967	1876.294359	1847.184924	1888.926953	1842.878175
	Standard value	127.2561383	91.22185049	69.00003268	32.76980351	140.693674	31.32108747
H22-7	Average value	2027.479588	2030.758499	2029.556604	2032.238674	2028.177978	2029.128603
	Standard value	6.106897592	8.027195324	5.81348717	7.446489204	8.003968446	8.197733191
H22-8	Average value	2223.108804	2223.537417	2222.070633	2223.140251	2220.888475	2220.690533
	Standard value	4.749655105	2.963408213	4.895282849	3.995669404	5.451654006	6.337353983
H22-9	Average value	2510.930321	2510.930321	2536.358938	2498.216012	2523.644629	2498.216012
	Standard value	65.93880108	65.93880108	85.778947	48.38585173	77.58997694	48.38585173
C22-10	Average value	2594.615905	2596.833927	2585.256107	2591.210109	2605.304194	2619.308989
	Standard value	48.2013289	49.71807546	57.1034079	56.36586785	42.57395199	34.10382553
C22-11	Average value	2695.981932	2685.587394	2733.855734	2710.621315	2700.168413	2715.332781
	Standard value	116.3652035	110.1475838	146.333679	118.5098748	113.7913849	109.3008673
C22-12	Average value	2857.067086	2858.742176	2854.959949	2861.414681	2859.407788	2860.718769
	Standard value	9.364347909	14.88960231	5.539104327	17.96133754	15.00545163	16.34731781

Table 3.2 presents the results of evaluating the reliability of decisions made by each optimization method for managing information flows in the DSS [45–64].

Table 3.2 Assessment of the effectiveness of the proposed management method by the criterion of information processing reliability

Function Name	Metric	Particle Swarm Algorithm	Ant Colony Algorithm	Black Widow Algorithm	Grey Wolf Swarm Algorithm	Canonical Brown Bear Algorithm	Improved Brown Bear Algorithm
1	2	3	4	5	6	7	8
U22-1	Average value	0.66	0.73	0.67	0.68	0.8	0.9
	Standard value	0.7	0.73	0.68	0.69	0.83	0.91
B22-2	Average value	0.7	0.73	0.7	0.71	0.77	0.89
	Standard value	0.71	0.73	0.72	0.72	0.76	0.9
B22-3	Average value	0.68	0.73	0.7	0.71	0.76	0.92
	Standard value	0.69	0.73	0.69	0.73	0.77	0.91
B22-4	Average value	0.67	0.74	0.7	0.72	0.78	0.93
	Standard value	0.67	0.72	0.67	0.72	0.79	0.92
B22-5	Average value	0.6	0.71	0.64	0.73	0.8	0.91
	Standard value	0.61	0.72	0.64	0.74	0.88	0.92
B22-6	Average value	0.64	0.73	0.66	0.77	0.85	0.93
	Standard value	0.66	0.75	0.66	0.78	0.83	0.92
H22-7	Average value	0.67	0.72	0.68	0.75	0.81	0.9
	Standard value	0.68	0.71	0.69	0.74	0.83	0.9
H22-8	Average value	0.68	0.74	0.69	0.75	0.84	0.93
	Standard value	0.65	0.74	0.67	0.77	0.81	0.91

Continuation of Table 3.2

1	2	3	4	5	6	7	8
H22-9	Average value	0.64	0.75	0.66	0.69	0.83	0.91
	Standard value	0.7	0.72	0.71	0.71	0.84	0.93
C22-10	Average value	0.69	0.71	0.7	0.72	0.8	0.94
	Standard value	0.68	0.71	0.7	0.73	0.8	0.91
C22-11	Average value	0.67	0.71	0.69	0.71	0.82	0.91
	Standard value	0.67	0.72	0.68	0.74	0.91	0.91
C22-12	Average value	0.63	0.73	0.65	0.75	0.82	0.91
	Standard value	0.62	0.74	0.66	0.76	0.83	0.91

From the analysis of **Tables 3.1** and **3.2**, it can be concluded that the proposed method ensures stable algorithm performance for the main test functions of unimodal and multimodal types.

As seen from **Tables 3.1–3.2**, an improvement in decision-making efficiency is achieved at the level of 15–18 % due to the use of additional procedures and ensuring decision reliability at the level of 0.9.

3.2 METHOD FOR EVALUATING THE EFFICIENCY OF PROCESSING DIVERSE DATA IN DECISION SUPPORT SYSTEMS

The most promising way to enhance the effectiveness of DSS functioning is to improve the efficiency of processing diverse data based on defined indicators (criteria). One approach to achieving this improvement is the use of methods based on artificial intelligence [56–66].

Given the specific features of DSS operation, which involve processing both structured and unstructured data and storing both processed and unprocessed datasets, the most appropriate approach is the use of bio-inspired algorithms.

Bio-inspired algorithms have found practical applications in solving real-world tasks, such as engineering calculations and processing large data arrays, as well as specialized tasks like assessing operational environments.

However, most of the basic bio-inspired algorithms fail to balance exploration and exploitation, leading to unsatisfactory performance in complex real-world optimization tasks [67–86].

This necessitates the introduction of various strategies to improve the convergence speed and accuracy of basic bio-inspired algorithms. One way to improve decision-making efficiency using bio-inspired algorithms is their combination, i.e., adding the basic procedures of one algorithm to another.

To achieve the research goal, the following interrelated research tasks need to be addressed:

– develop an algorithm for implementing the method;

– evaluate the efficiency of the proposed method.

The aim of the research is to develop a method to improve decision-making efficiency in DSS.

This section of the study proposes an approach for evaluating the efficiency of processing diverse data in DSS based on the simulation of snow ablation. This approach assumes that when the DSS transitions from one operating mode to another, there is a significant increase in the amount of information circulating in the decision support systems. At this stage, a substantial increase occurs in the amount of information that needs to be processed, and the quality of this processing must be evaluated based on defined performance indicators:

Action 1. Initialization of the initial population.

The operation of the snow ablation algorithm (SAA) begins with the formation of the initial population, which is generated randomly. Depending on the number of information sources additionally included in the operation (transitioning to another operating mode), the number of search agents in the population is determined.

The entire population is represented as a matrix with columns *Dim* and *N* rows, where *N* – the size of the swarm, and *Dim* – the number of dimensions in the decision space for the population agents (3.11):

$$Z = L + \left(\theta \times \left(U - L\right)\right)\iota = \begin{bmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,Dim-1} & z_{1,Dim} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,Dim-1} & z_{2,Dim} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ z_{N-1,1} & z_{N-1,2} & \cdots & z_{N-1,Dim-1} & z_{N-1,Dim} \\ z_{N,1} & z_{N,2} & \cdots & z_{N,Dim-1} & z_{N,Dim} \end{bmatrix}_{N\times Dim}, \tag{3.11}$$

where *L* and *U* – the lower and upper bounds of the solution space, respectively; θ – a randomly generated number in the range [0, 1]; ι – the type of uncertainty in the data circulating in decision support systems (DSS).

Action 2. Numbering the agents in the population, i, $i \in [0, S]$.

Each agent in the population is assigned a sequential number.

Action 3. Determining the initial velocity of the agents in the population.

The initial velocity v_0 of each agent in the population is determined by the following expression (3.2).

In the proposed approach, the position of each population agent in the problem-solving space is updated based on the modeling of exploration and exploitation strategies.

Exploration Phase.

Action 4. Preliminary evaluation of the search plane by the population agents.

In this procedure, the search area is conceptually defined as the snow ablation plane.

Action 5. Identifying the weakest points of the snow cover.

The number of weak points in the snow cover significantly depends on the environmental temperature (T) and the intensity of solar radiation. These are calculated using the following equations (3.12), (3.13):

$$T = \exp\left(\frac{-t}{t_{\max}}\right), \tag{3.12}$$

$$FQ = c_1 \exp\left(\frac{t - t_{\max}}{t_{\max}}\right), \tag{3.13}$$

where t – the current iteration number; t_{max} – the total number of iterations; c_1 – the coefficient for snow cover melting within specific temperature ranges.

In this algorithm, agents select the weakest points in the snow cover according to c_1, then update their positions.

For DSS, this procedure is designed to identify the most loaded information flows.

Action 6. Movement of agents on the search plane.

Due to the non-linear movement, the search agents exhibit high decentralization, particularly when snow or melted snow turns into vapor. This process is described using Brownian motion. For typical Brownian motion, the step size is determined by the probability density function based on a normal distribution with zero mean and unit variance:

$$f_{BM}\left(x;0,1\right) = \frac{1}{\sqrt{2\pi}} \times \exp\left(-\frac{x^2}{2}\right). \tag{3.14}$$

The formula for determining the positions of the search agents throughout the exploration process is as follows:

$$Z_i\left(t+1\right) = Elite\left(t\right) + BM_i\left(t\right) \otimes \left(\theta_1 \times \left(G\left(t\right) - Z_i\left(t\right)\right) + \left(1-\theta_1\right) \times \left(\bar{Z}\left(t\right) - Z_i\left(t\right)\right)\right), \tag{3.15}$$

where $\otimes$ – denotes element wise multiplication; θ_1 – random numbers in the range [0, 1]; $Z_i(t)$ – i-th h agent at iteration t-th iteration; $BM_i(t)$ – a vector including random values based on the Gaussian distribution representing Brownian motion; $\bar{Z}\left(t\right)$ – the centroid position of the population; $Elite(t)$ – a randomly selected member of the elite subset in the population; $G(t)$ – the current best solution.

Below are the corresponding mathematical expressions describing the variables in expression (3.15):

$$\bar{Z}\left(t\right) = \frac{1}{N}\sum_{i=1}^{N} Z_i\left(t\right). \tag{3.16}$$

CHAPTER 3

$$Elite(t) \in \left[G(t), Z_{second}(t), Z_{third}(t), Z_c(t)\right], \tag{3.17}$$

where $Z_{third}(t)$ and $Z_{second}(t)$ – the third and second best individuals in the current population.

The position of the centroid for those individuals in the population whose metrics fall within the top 50 % is denoted as $Z_c(t)$.

$Z_c(t)$ is calculated using equation (3.18):

$$Z_c(t) = \frac{1}{N_1} \sum_{i=1}^{N_l} Z_i(t), \tag{3.18}$$

where $Z_i(t)$ – i-th best leader; N_l – the number of leaders in the population.

As a result, the elite is selected randomly from a set that includes the centroid positions of the leaders, the current best solution, and the second and third best individuals during each iteration.

Exploitation Phase.

Action 7. Searching for solutions by population agents on the search plane.

The degree-day method, one of the most used models for describing snow melting, is applied in this study to simulate the snow ablation process:

$$M = DDF \times (T - T_1), \tag{3.19}$$

where M – the snow melting rate; T – the average daily temperature; $T_1 = 0$.

Accordingly, M is calculated using the following mathematical expression:

$$M = DDF \times T, \tag{3.20}$$

where DDF – a coefficient ranging from 0.35 to 0.6. The update of DDF at each iteration is described by the following expression:

$$DDF = 0.35 + 0.25 \times \frac{e^{\frac{t}{t_{max}}} - 1}{e - 1}, \tag{3.21}$$

where t_{max} – the stopping condition for the algorithm.

The snow melting rate is then calculated using the formula:

$$M = \left(0.35 + 0.25 \times \frac{e^{\frac{t}{t_{max}}} - 1}{e - 1}\right) \times T(t), T(t) = e^{\frac{-t}{t_{max}}}. \tag{3.22}$$

Action 8. Updating the positions of population agents on the search plane.

The equation for updating the positions of agents on the search plane during the exploitation phase of the Snow Ablation Algorithm (SAA) is as follows:

$$Z_i(t+1) = M \times G(t) + BM_i(t) \otimes \left(\theta_2 \times \left(G(t) - Z_i(t)\right) + (1-\theta_2) \times \left(\bar{Z}(t) - Z_i(t)\right)\right), \qquad (3.23)$$

where θ_2 – a random integer in the range [–1, 1]; M – the snow melting rate.

Action 9. Training knowledge bases.

In this study, a method of training knowledge bases for each population agent is employed based on evolving artificial neural networks, as proposed in [2]. This method is used to modify the movement behavior of each population agent to achieve more accurate analysis results in subsequent iterations.

Action 10. Determining the required computational resources for the decision support system.

To avoid computational loops in *Actions 1–10* of the method and to improve the efficiency of computations, the load of decision support systems is additionally assessed. If the predefined threshold of computational complexity is exceeded, the required number of additional software and hardware resources is determined using the method proposed in [31].

End of the Algorithm.

The efficiency of the proposed algorithm is analyzed based on the decision-making efficiency criterion presented in **Table 3.3** [60–86].

Table 3.3 Comparative analysis of algorithms based on the decision-making efficiency criterion

Algorithm Name	Average value	Average value	Worst solution	Standard deviation	Median value
Improved Snow Ablation Algorithm	0.012672	0.012701	0.012706	0.001106	0.012700
White Shark Algorithm	0.012722	0.012754	0.012766	0.007391	0.012744
Tree Seed Algorithm	0.012782	0.012799	0.01283	0.00567	0.012802
Bee Colony Algorithm	0.012786	0.012812	0.012836	0.004191	0.012815
Penguin Swarm Algorithm	0.013305	0.014951	0.018023	0.002293	0.013312
Grey Wolf Swarm Method	0.012926	0.014594	0.018	0.001636	0.014147
Canonical Snow Ablation Algorithm	0.012983	0.01356	0.01434	0.000289	0.013488
Particle Swarm Optimization Method	0.013147	0.014162	0.016398	0.002092	0.013119
Genetic Algorithm	0.012885	0.013188	0.015352	0.000378	0.013069

The results of the simulation presented in **Table 3.3** indicate an improvement in the efficiency of processing diverse data by 13–17 % due to the use of additional enhanced procedures.

3.3 METHOD OF EVALUATION AND FORECASTING IN INTELLIGENT DECISION SUPPORT SYSTEMS

To enable the analysis of the state of a monitored object and ensure its forecasting, a systematic approach to the analysis and forecasting of its condition is proposed.

Fig. 3.1 shows the structural diagram of the system for managing the process of analyzing and forecasting the state of an object, which is divided into [11, 30]:

1) control subsystem (control subject, *S*);

2) managed subsystem (control object, *O*);

3) object model (in this case, a fuzzy cognitive model *Y*). The fuzzy cognitive model is used because the state of the analyzed object is typically characterized by both quantitative and qualitative indicators. This requires converting them to a unified measurement scale.

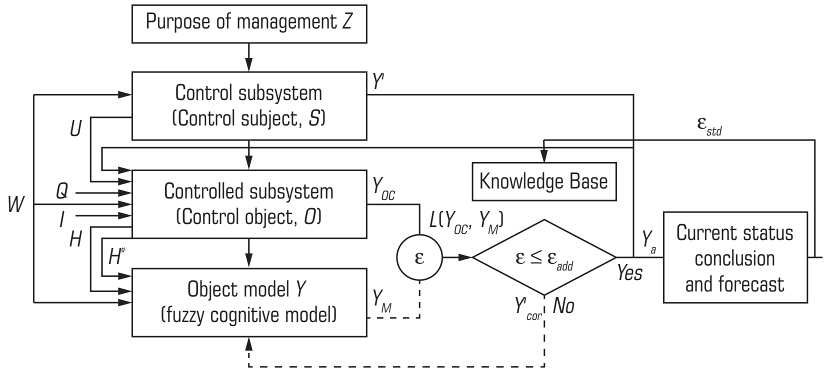

Fig. 3.1 Structural diagram of the system for analyzing and forecasting the state of an object

Explanation of variables from **Fig.3.1**:

W – external information;

Q – system resources required for analyzing and forecasting the state of the object;

H – internal information necessary for building fuzzy cognitive models (FCMs);

H^* – corrected error;

U – control impact (management decisions, control commands) (direct connection);

Y_{OC} – output information (actual data, parameters, indicators) characterizing the state of the control object;

Y_M – model output parameters (desired or expected parameters);

ε – error (discrepancy);

ε_{thr} – predefined fixed threshold value;

$L(Y_{OC}, Y_M)$ – validation of the correspondence between the data obtained from the model and the real object it describes;

Y' – information about the state of the object (feedback);

Y'_{cor} – model correction (addition of new factors and relationships);

Y_a – adequate model of the monitored object corresponding to its real state;

ε_{std} – knowledge base update.

In the managed subsystem (O), the control objects are considered the targets of management impacts. The object model refers to the development and study of a fuzzy cognitive model (FCM) for assessing the state of the object using the methodology of fuzzy cognitive modeling.

The control subsystem generates the control impact U based on management objectives and information received from the external environment W.

The managed subsystem receives information (Q, I, U) that forms tasks for analyzing and forecasting the state of the object.

Using W, Q, I, fuzzy cognitive models are developed and studied through fuzzy cognitive modeling methodology. These models enable the exploration and analysis of possible development scenarios for the objects. The development scenarios refer to the evolution of situations related to the monitored object's behavior.

If the obtained results (calculated values) Y_M do not match the actual results characterizing the state Y_{OC} (the condition $\varepsilon \le \varepsilon_{per}$ is not met), the control subsystem adjusts the FCM (Y_{cor}). If $\varepsilon \le \varepsilon_{per}$ is satisfied, the FCM is deemed adequate Y_a. An adequate FCM allows the prediction of the object's behavior.

To validate the adequacy of the model, a "historical method" is proposed. This method involves applying the developed FCMs to similar past situations with known dynamics. If the obtained results align with the real course of events, the FCM is considered functional and valid.

Control is executed using feedback Y'. The control subsystem receives feedback Y', from the managed subsystem and the external environment W. It processes this information, compares it with the desired characteristics of the control object, and makes a new decision, generating the next control impact U based on it. The managed subsystem also processes feedback Y', compares it with the desired characteristics of the control object, and corrects the error H^*.

The system for managing the process of analyzing and forecasting the state of objects can be represented as a tuple:

$$S_c = \langle S, O, Y, Z, W, Q, Y_a, D \rangle, \tag{3.24}$$

where Z – the management goal; $D = \langle I, H, U, Y_{OC}, Y_M, Y', H^*, Y'_{cor} \rangle$ – the internal environment of the management system S_c; $Y = \langle W, H, H^*, Y_M \rangle$ – the object model, with the result Y_M being the FCM.

Let's record expression (3.24) for the dynamic system:

$$\forall t \in \{1,\ldots,T,\ldots\}\, S_t = \begin{cases} s_1^{(t)} F_1\left(\varphi_{1,1}\left(s_1^{(t-1)},\ldots,s_1^{(t-L_1^1)}\right),\varphi_{1,N}\left(s_N^{(t-1)},\ldots,s_N^{(t-L_1^N)}\right)\right)\times \iota_1, \\ s_2^{(t)} F_2\left(\varphi_{2,1}\left(s_1^{(t-1)},\ldots,s_1^{(t-L_2^1)}\right),\varphi_{2,N}\left(s_N^{(t-1)},\ldots,s_N^{(t-L_2^N)}\right)\right)\times \iota_2, \\ \ldots \\ s_N^{(t)} F_N\left(\varphi_{N,1}\left(s_1^{(t-1)},\ldots,s_1^{(t-L_N^1)}\right),\varphi_{N,N}\left(s_N^{(t-1)},\ldots,s_N^{(t-L_N^N)}\right)\right)\times \iota_N, \end{cases} \tag{3.25}$$

where S – a multidimensional time series; $S_t = \left(s_1^{(t)}, s_2^{(t)},\ldots,s_N^{(t)}\right)$ – the time slice of the object's state, represented as a multidimensional time series at moment t; $s_j^{(t)}$ – the value of the j-th component of the multidimensional time series at moment t; L_j^i – the maximum time delay of the i-th component relative to the j-th; ϕ_{ij} – the maximum time delay of the i-th component relative to the j-th; F_i – the transformation used to obtain $s^{(t)}$, $i = 1\ldots, N$; N – the number of components in the multidimensional time series; ι – the operator accounting for the degree of awareness about the object's state.

From expression (3.25), it can be concluded that this formula describes the processes within the analyzed object while considering time delays. These delays are necessary for collecting, processing, and summarizing information, as well as accounting for the degree of awareness about the object's state. Additionally, expression (3.25) allows for the description of processes with both quantitative and qualitative units of measurement, as well as processes depicted in **Fig. 3.1**.

The method of evaluation and forecasting in intelligent decision support systems consists of the following sequence of actions (**Fig. 3.2**):

1. Input of initial data.

At this stage, the initial data available about the object to be analyzed is entered. The base model of the object's state is initialized.

2. Identification of factors and relationships between them.

In existing studies, such as [3, 8, 13], the stage of processing initial data and the initial uncertainty of the type of information to be modeled is not considered. For simplification, the authors often assume that factor values are represented as dimensionless quantities within the interval [0, 1], and the values of relationships between them are within the interval [–1, 1]. To address this issue, a procedure for processing uncertain initial data during the identification of factors and their relationships is proposed.

Action 2.1. Input of Initial Data (Values of the parameters of FCM nodes, relationships between them, and the a priori type of uncertainty in the initial data).

The a priori types of uncertainty in the initial data can be complete uncertainty, partial uncertainty, or complete awareness. The parameter values of the nodes x_{v_i}, $i = \overline{1,h}$ (h – number of factors) can be represented as:

1. Numbers that differ in units of measurement, magnitude, and verbal descriptions.
2. Intervals, fuzzy triangular numbers, fuzzy trapezoidal numbers, or fuzzy polyhedral numbers.

The initial parameter values of the nodes are simultaneously represented in each of the listed forms, while the initial values of the relationships between them are represented only in one of these forms.

Action 2.2. Conditional Check:

– if the parameter values of the nodes are represented as intervals or fuzzy numbers (e.g., intervals, fuzzy triangular numbers, fuzzy trapezoidal numbers, or fuzzy polyhedral numbers), then proceed to *Action 2.3*. If this condition is not met, proceed to *Action 2.4*.

Action 2.3. Normalization of Node Parameter Values Represented as Intervals and Fuzzy Numbers.

As a result of normalization, the parameter values of the nodes are represented as intervals with normalized parameter values. To obtain a single normalized fuzzy value from an interval, the following is recommended:

– for normalized intervals, fuzzy trapezoidal numbers, and fuzzy polyhedral numbers, select the arithmetic mean;

– for normalized fuzzy triangular numbers, select the expected value for normalization.

Action 2.4. Conditional Check:

– if the condition that the parameter values of the nodes are represented as verbal descriptions is satisfied, proceed to *Action 2.5*;

– if the condition is not satisfied, proceed to *Action 2.6*.

Action 2.5. Structuring Node Parameter Values.

After completing this action, proceed to *Action 2.8*.

For node parameters represented as verbal descriptions, it is proposed to perform structuring, where each verbal description of the node parameter is assigned a corresponding numerical value from the interval [0, 1]. To evaluate the node parameter values, a verbal description "Factor Level" is introduced (**Table 3.4**).

CHAPTER 3

Table 3.4 Evaluation of node parameter values for the verbal variable "Factor Level"

Verbal description	Numerical value
Low	[0.1, 0.3]
Below average	[0.31, 0.5]
Average	[0.51, 0.7]
Above average	[0.71, 0.9]
High	[0.91, 1]

Normalization and structuring of node parameter values are necessary to ensure that the numerical values of the node parameters do not differ in units of measurement, orders of magnitude, and belong to the interval [0, 1].

Action 2.6. Conditional Check:

– if the condition that the parameter values of the nodes are represented as numbers (not differing in units of measurement and orders of magnitude) is satisfied, proceed to *Action 2.8*;

– if the condition is not satisfied (i.e., the parameter values of the nodes differ in units of measurement and orders of magnitude), proceed to *Action 2.7*.

Action 2.7. Normalization of Node Parameter Values Represented as Numbers:

$$x_{v_i}^{\text{norm}} = \frac{x_{v_{i\ \text{nes}}} - x_{v_{i\ \text{min}}}}{x_{v_{i\ \text{max}}} - x_{v_{i\ \text{min}}}}, x_{v_i}^{\text{norm}} \in [0,1], \tag{3.26}$$

where $x_{v_{i\ \text{nes}}}$ – the current value of the node parameter; $x_{v_{i\ \text{max}}}$, $x_{v_{i\ \text{min}}}$ – the minimum and maximum values of the node parameter, respectively $v_i \in V$, $i = \overline{1,h}$.

Formula (3.3) for normalizing the values of node parameters represented as intervals, fuzzy triangular, and fuzzy trapezoidal (polyhedral) numbers is not suitable. This is because the interval values of the node parameters x_{v_i} must not overlap, as only in this case can the relationships "greater than" (maximum) or "less than" (minimum) be established. For intervals $a = [a_1, a_2]$ and $b = [b_1, b_2]$ to be comparable in terms of $a \geq b$, it is necessary and sufficient that the condition $a_1 \geq b_1$, $a_2 > b_2$.

Action 2.8. Normalization of Relationship Values Between Nodes Represented as Intervals or Fuzzy Numbers.

The nature and strength of relationships between nodes, represented as intervals, fuzzy triangular, and fuzzy trapezoidal (polyhedral) numbers, are evaluated on a five-point scale, as shown in **Table 3.5**.

Table 3.5 Evaluation of the nature and strength of relationships between nodes

Numerical value	**Verbal description**
1	**2**
For intervals	
0	Absent
[0.1, 1] [–0,1, –1]	Very weakly strengthens Very weakly weakens
[1.1, 2] [–1.1, –2]	Weakly strengthens Weakly weakens
[2.1, 3] [–2.1, –3]	Moderately strengthens Moderately weakens
[3.1, 4] [–3.1, –4]	Strongly strengthens Strongly weakens
[4.1, 5] [–4.1, –5]	Very strongly strengthens Very strongly weakens
For fuzzy triangular numbers	
0	Absent
[0.1, 0.5, 1] [–0.1, –0.5, –1]	Very weakly strengthens Very weakly weakens

Continuation of Table 3.5

1	2
[1.1, 1.5, 2] [–1.1, –1.5, –2]	Weakly strengthens Weakly weakens
[2.1, 2.5, 3] [–2.1, –2.5, –3]	Moderately strengthens Moderately weakens
[3.1, 3.5, 4] [–3.1, –3.5, –4]	Strongly strengthens Strongly weakens
[4.1, 4.5, 5] [–4.1, –4.5, –5]	Very strongly strengthens Very strongly weakens
For fuzzy trapezoidal numbers	
0	Absent
[0.1, 0.3, 0.6, 1] [–0.1, –0.3, –0.6, –1]	Very weakly strengthens Very weakly weakens
[1.1, 1.3, 1.6, 2] [–1.1, –1.3, –1.6, –2]	Weakly strengthens Weakly weakens
[2.1, 2.3, 2.6, 3] [–2.1, –2.3, –2.6, –3]	Moderately strengthens Moderately weakens
[3.1, 3.3, 3.6, 4] [–3.1, –3.3, –3.6, –4]	Strongly strengthens Strongly weakens
[4.1, 4.3, 4.6, 5] [–4.1, –4.3, –4.6, –5]	Very strongly strengthens Very strongly weakens
For fuzzy polyhedral numbers	
0	Absent
[0.1, w_{ijn} / N, 1] [–0.1, w_{ijn} / N, –1]	Very weakly strengthens Very weakly weakens
[1.1, w_{ijn} / N, 2] [–1.1, $-w_{ijn}$ / N, –2]	Weakly strengthens Weakly weakens
[2.1, w_{ijn} / N, 3] [–2.1, w_{ijn} / N, –3]	Moderately strengthens Moderately weakens
[3.1, w_{ijn} / N, 4] [–3.1, w_{ijn} / N, –4]	Strongly strengthens Strongly weakens
[4.1, w_{ijn} / N, 5] [–4.1, w_{ijn} / N, –5]	Very strongly strengthens Very strongly weakens

As a result of normalization, the values of the relationships between nodes are represented as intervals with normalized relationship values. To obtain a single normalized fuzzy value from an interval, the following is recommended:

1) for normalized intervals $w_{ij}^{norm}=\left[w_{ij1}^{norm},w_{ij2}^{norm}\right]$, fuzzy trapezoidal numbers $w_{ij}^{norm}=\left[w_{ij1}^{norm},w_{ij2}^{norm},w_{ij3}^{norm},w_{ij4}^{norm}\right]$ and fuzzy polyhedral numbers $w_{ij}^{norm}=\left[w_{ij1}^{norm},\ldots,w_{ijN}^{norm}\right]$ select the arithmetic mean $w_{ij}^{*norm}=\dfrac{w_{ijn}^{norm}}{N}$;

2) for normalized fuzzy triangular numbers $w_{ij}^{norm}=\left[w_{ij1}^{norm},w_{ij2}^{norm},w_{ij3}^{norm}\right]$ elect the expected normalized value $w_{ij1}^{*norm}=w_{ij2}^{norm}$, where w_{ij}^{norm} – normalized interval values of the relationships between nodes v_i and v_j $w_{ij}^{norm}\in\left[-1,1\right]$; w^{*max}.

Action 2.9. Structuring the Values of Relationships Between Nodes.

To establish cause-and-effect relationships, a scale has been defined to evaluate the nature and strength of relationships between nodes (**Table 3.6**).

Structuring involves the following: each value of a relationship, represented as a verbal description, is assigned a corresponding numerical value from the interval [–1, 1].

Table 3.6 Evaluation of the nature and strength of relationships between nodes represented as verbal descriptions

Verbal description	**Numerical value**
Absent	0
Very weakly strengthens Very weakens	[0.1, 0.3] [–0.1, –0.3]
Weakly strengthens Weakly weakens	[0.31, 0.5] [–0.31, –0.5]
Moderately strengthens Moderately weakens	[0.51, 0.7] [–0.51, –0.7]
Strongly strengthens Strongly weakens	[0.91, 1] [–0.91, –1]

Normalization and structuring of the values of relationships between nodes are necessary to ensure that all relationship values belong to the interval [–1, 1].

3. Construction of the FCM. Formation of the Structure (Preliminary Structural Adjustment).

The formation of the FCM involves setting structural interconnections (represented as temporal lags) between the concepts of the FCM, weighted by fuzzy values $w_{ij}^{\left(t-l_i^j\right)}$ of their influence on each other. In this study, the FCM FS_i, which implements fuzzy temporal transformations F_i, is proposed as modified ANFIS-type models (Adaptive Neuro-Fuzzy Inference System). The FCM ensures the formation, storage, and output of predicted fuzzy values for the corresponding components of the multidimensional time series with the temporal delays required by the FCM.

The input fuzzy temporal variables of the FS_i model for concept C_i are associated with the output fuzzy temporal variables of those concepts that directly influence concept C_i. At the same time, the input fuzzy temporal variables of C_i are preliminarily "weighted" by the corresponding fuzzy degrees of influence $w_{ij}^{\left(t-l_i^j\right)}$, based on which the following transformation is performed:

$$\tilde{s}_j^{'(t-l_i^j)} = \left(w_{ij}^{(t-l_i^j)} \mathrm{T}\, \tilde{s}_j^{(t-l_i^j)} \right), l_i^j = 0,\dots,L_i^j, \tag{3.27}$$

where T – the T-norm operation.

The output fuzzy temporal variables of the FS_i model for concept C_i are designed for the formation, storage, and output of predicted values for the i-th component of the multidimensional time series, corresponding to the temporal lags. To construct the fuzzy component temporal models FS_i both a priori information about the components of the multidimensional time series available in the knowledge base and data obtained through evaluation or measurements can be used.

In the first case, it is assumed that the task of ensuring the completeness and consistency of the fuzzy rule base for the FS_i model has been solved in advance.

If only experimental data are available, the task becomes one of model identification. In practice, the mixed case is most common, where the initial rule base of the model is built on heuristic assumptions, and its parametric adjustment (training) is performed based on a training dataset.

The input fuzzy temporal variables of the FS_i model are $S_1^{'} = \left\{ \tilde{s}_3^{'(t-1)}, \tilde{s}_3^{'(t-3)}, \tilde{s}_4^{'(t-3)}, \tilde{s}_5^{'(t-3)}, \tilde{s}_1^{'(t-3)} \right\}$ while the output fuzzy temporal variables are $S_1^{'} = \left\{ \tilde{s}_1^{(t)}, \tilde{s}_1^{'(t-1)}, \tilde{s}_1^{'(t-2)} \right\}$.

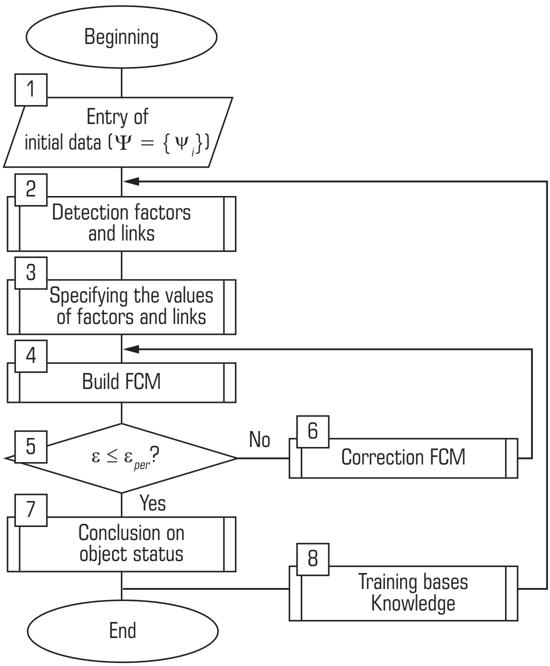

Fig. 3.2 Algorithm for implementing the method of object state analysis and forecasting

When constructing the model, the degrees of truth are first determined for the current values of the input variables in terms of their correspondence to the fuzzy premises of all model rules. Subsequently, aggregation is performed based on the *T*-norm operation for the degrees of truth of the premises of the rules:

$$\alpha_p = \min \mu_{\tilde{L}}\left(\tilde{s}_1^{'(t-1)}\right), \mu_{\tilde{L}}\left(\tilde{s}_3^{'(t-3)}\right), \mu_{\tilde{M}}\left(\tilde{s}_4^{'(t-3)}\right), \mu_{\tilde{M}}\left(\tilde{s}_5^{'(t-3)}\right), \mu_{\tilde{H}}\left(\tilde{s}_1^{'(t-3)}\right). \tag{3.28}$$

Next, the conclusions of the corresponding rules are activated according to the degrees of truth of their premises, using the implication operation (in this case, the Mamdani implication – the min-activation operation):

$$\mu_{\tilde{M}}\left(\tilde{s}_1^{'(t)}\right) = \min\left(\alpha_p, \tilde{M}\right). \tag{3.29}$$

Next, the max-disjunction operation is performed, accumulating the activated conclusions of all the model rules:

$$\tilde{s}_1^{(t)} = \max\left(\mu_{\tilde{M}}\left(\tilde{s}_1^{'(t)}\right),\ldots,\mu_{\tilde{M}}\left(\tilde{s}_1^{'(t)}\right),\ldots,\mu_{\tilde{H}}\left(\tilde{s}_1^{'(t)}\right)\right). \tag{3.30}$$

After that, normalization, storage, and output of fuzzy values of the model's output variables are performed with the required temporal delays for the FCM:

$$\tilde{s}_{1(norm)}^{(t)} = Z^0\left(\tilde{s}_1^{(t-1)}\right), \tilde{s}_{1(norm)}^{(t-2)} = Z^{-1}\left(\tilde{s}_1^{(t-1)}\right). \tag{3.31}$$

4. Training Artificial Neural Networks (ANNs).

In this procedure, ANNs are trained using the method for training evolving ANNs developed by the authors in [2]. This method differs from existing ones in that it allows training not only synaptic weights but also the parameters of membership functions along with the ANN architecture. At this stage, all fuzzy component temporal models of the FCM are also aligned. The alignment of all fuzzy component temporal models FS_i, i = 1,…,N of the FCM is carried out after their "personalized" parametric adjustment. Alignment involves modifying the modal values and degrees of fuzziness of the fuzzy degrees of influence $\left\{ w_{ij}^{\left(t-l_i^j\right)} \middle| l_i^j = 0,\ldots,L_i^j \right\}$ between the FCM concepts to ensure maximum improvement in the prediction accuracy of each component of the multidimensional time series without deterioration. Before aligning the fuzzy component temporal models of the FCM, an additional "alignment" training dataset is formed, consisting of retrospective data for all components of the multidimensional time series simultaneously. The alignment procedure for all fuzzy component temporal models of the FCM is considered successfully completed if the total error for each of these models does not exceed a predefined threshold. For well-aligned components of the multidimensional time series, or for these models, the Edgeworth-Pareto principle will be applied.

5. Forecasting the State of the Analyzed Object.

Multidimensional analysis and forecasting of the state of a complex system/process are carried out based on a structurally and parametrically adjusted FCM and can be performed in the following modes:

– firstly, direct multidimensional forecasting of the state of a complex system/process for the t-th moment in time, i.e., the calculation of output variable values of the FS_i, $i = 1,\ldots,N$ for the corresponding sets of input variable values of these models, given each time;

– secondly, self-development and predictive assessment of changes in the state of a complex system/process, where the modeling of state dynamics is conducted starting from a certain situation defined by the initial values of all FCM concepts, in the absence of external influences on it;

– thirdly, development and predictive assessment of changes in the state of a complex system/process, where the modeling of state dynamics is conducted starting from a certain situation. The situation is defined by the initial values of all FCM concepts, under external influence on the values of concepts and/or on the relationships between the concepts of the FCM.

The proposed method of evaluation and forecasting in intelligent decision support systems. To assess the effectiveness of the developed method of evaluation and forecasting, a comparative assessment was performed with the most popular software products:

– ARIS Business Performance Edition (IDS Scheer AG, Germany);

– IBM WebSphere Business Modeler (IBM, USA);

– System21 Aurora (Campbell Lee Computer Services Limited, UK);

– SAP Strategic Enterprise Management (SAP, Germany);

– Hyperion Performance Scorecard (Oracle, USA);

– CA ERWin Process Modeler (CA, USA).

The modeling of the method for decision search processing was conducted according to the algorithm in **Fig. 3.2** and expressions (3.24)–(3.31). The proposed method for evaluation and forecasting was modeled in the MathCad 14 software environment (USA). The task solved during the modeling was the evaluation of elements of the operational environment of a grouping of troops (forces) (**Table 3.7**).

Table 3.7 Incidence matrix of the cognitive map for situation evaluation

No.	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}	w_{12}
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
w_1	0	1	1	0	0	0	0	1	0	1	1	0
w_2	0	0	1	0	1	1	1	0	0	1	1	0
w_3	0	1	0	0	1	0	0	–1	0	1	0	–1
w_4	0	0	1	0	0	1	–1	0	0	1	1	0
w_5	0	1	1	0	0	0	0	1	1	1	1	0
w_6	0	1	0	0	–1	0	1	1	–1	1	1	0
w_7	1	–1	1	0	0	–1	0	1	0	1	0	0
w_8	0	–1	1	1	1	–1	0	0	0	0	0	0

Continuation of Table 3.7

	1	2	3	4	5	6	7	8	9	10	11	12
w_9	1	0	1	1	–1	1	1	0	0	1	1	0
w_{10}	1	–1	0	1	0	1	0	–1	0	0	0	0
w_{11}	1	1	1	–1	0	1	0	0	0	1	1	1
w_{12}	0	0	1	1	0	1	1	1	1	1	0	0

The results of the assessment of the operational environment of the grouping based on the input data are presented in **Table 3.8**, which provides the normalized evaluation results.

Table 3.8 Comparison of computational complexity between software and the developed method for operational environment assessment

No.	Software name	Number of calculations	Developed method (by number of calculations)
1	ARIS Business Performance Edition (IDS Scheer AG)	67000	58960
2	IBM WebSphere Business Modeler (IBM)	64500	58760
3	System21 Aurora (Campbell Lee Computer Services Limited)	57000	48450
4	SAP Strategic Enterprise Management (SAP)	39830	35847
5	Hyperion Performance Scorecard (Oracle)	46200	40194
6	CA ERWin Process Modeler (CA)	43050	37023

From the analysis of the data presented in **Table 3.8**, it is evident that the proposed method requires fewer calculations compared to known approaches for evaluation and forecasting. The advantage of the proposed method, compared to existing ones, lies in the reduction of computational complexity, which in turn increases the decision-making efficiency regarding the operational environment of troop (force) groupings. **Tables 3.9** and **3.10** present comparative results of the efficiency of training evolving artificial neural networks.

Table 3.9 Comparative results of the efficiency of training evolving artificial neural Networks

System	Algorithm parameters	XB (Xie-Beni Index)	Time, sec
FCM (Fuzzy C-Means)	–	0.1903	2.69
EFCM	Dthr = 0.24	0.1136	0.14
EFCM	Dthr = 0.19	0.1548	0.19
Proposed System (batch mode)	delta = 0.1	0.0978	0.37
Proposed System (online mode)	delta = 0.1	0.1127	0.25

Before training, observation features were normalized to the interval [0, 1].

Table 3.10 Comparative results of clustering

System	Algorithm parameters	XB (Xie-Beni Index)	Time, sec
FCM (Fuzzy C-Means)	Dthr = 0.6	0.2963	0.81
EFCM	Dthr = 0.6	0.2330	0.54
Proposed System (batch mode)	delta = 0.4	0.2078	0.45
Proposed System (online mode)	delta = 0.4	0.2200	0.30

It is worth noting that the proposed training procedure demonstrated better results in terms of the partition coefficient (PC) compared to EFCM and better performance in terms of time compared to FCM. The study showed that the proposed training procedure provides, on average, 10–18 % higher training efficiency for artificial neural networks and does not accumulate errors during training (**Tables 3.9** and **3.10**).

These results can be observed from the last rows of **Tables 3.9** and **3.10**, as the difference in the Xie-Beni Index. Furthermore, as already mentioned, during their operation, known methods accumulate errors. For this reason, the proposed methodology incorporates the use of evolving artificial neural networks.

CONCLUSIONS

1. An algorithm for implementing the method of information flow management in intelligent decision support systems (IDSS) using a population-based algorithm was developed. Thanks to additional and improved procedures, the method allows:

– determining the initial population of ABC individuals and their initial positions on the search plane, taking into account the uncertainty of initial data on information flows in IDSS;

– considering the initial velocity of each ABC individual, which enables prioritization of the search in the corresponding search plane;

– providing universality in food-search strategies of ABC individuals, enabling classification of the conditions and factors influencing the process of information flow management in IDSS;

– exploring solution spaces for functions described by atypical functions through the use of ABC movement technique selection procedures;

– replacing unfit individuals by updating the ABC population;

– simultaneously searching for solutions in different directions;

– deep learning of ABC knowledge bases;

– calculating the necessary computational resources required if the current resources are insufficient for calculations.

2. A case study of the proposed method demonstrated a 15–18 % improvement in decision-making efficiency due to additional procedures and ensuring decision accuracy at a level of 0.9.

3. An algorithm for implementing the method of evaluating the efficiency of heterogeneous data processing in decision support systems was developed. Thanks to additional and improved procedures, the method allows:

– accounting for the type of uncertainty of data circulating in decision support systems;

– implementing adaptive strategies for search-plane exploration by population agents;

– considering the available computational resources of the subsystem for heterogeneous data processing in decision support systems;

– changing the search area for individual agents in the population;

– adjusting the movement speed of population agents;

– prioritizing the search by population agents;

– initializing the population based on the type of uncertainty;

– acting as a universal tool for analyzing the efficiency of heterogeneous data processing in decision support systems;

– verifying the adequacy of the obtained results;

– avoiding the problem of local extrema.

4. Simulation showed a 13–17 % improvement in data processing efficiency through additional refined procedures for introducing corrective coefficients related to data uncertainty.

5. A method for evaluation and forecasting in intelligent decision support systems was proposed.

The novelty of the proposed method includes:

– the use of a new type of fuzzy cognitive temporal models (FCTM) designed for multidimensional analysis and forecasting of object states under uncertainty;

– FCTM concepts are connected by subsets of fuzzy degrees of influence ordered in chronological sequence, accounting for the time lags of the corresponding components of the multidimensional time series;

– an improved object state forecasting procedure based on the new FCTM type, enabling multidimensional analysis, consideration of mediated influence, and interaction of components of the multidimensional time series with varying time lags relative to each other. It also ensures forecasting under conditions of non-stochastic uncertainty, nonlinearity, partial inconsistency, and significant interdependence of the multidimensional time series components;

– an enhanced training procedure for artificial neural networks (ANNs) in intelligent decision support systems, improving information processing efficiency and reducing errors by;

– training not only the synaptic weights of the ANN but also the type and parameters of the membership functions;

– training the architecture of the ANN;

– enabling the combination of ANN elements;

– enabling training of individual ANN elements;

– processing data in a single epoch without requiring storage of previous computations;

– avoiding error accumulation in ANN training as a result of processing incoming information.

A case study using the proposed method for forecasting the time series of a reconnaissance object demonstrated a 15–25 % improvement in ANN performance efficiency in terms of information processing due to additional refined procedures.

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 Computation and Natural Computation, 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, 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. et al. (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. Waterbird, 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
72. Mamoori, G. A., Sova, O., Zhuk, O., Repilo, I., Melnyk, B., Sus, S. et al. (2023). The development of solution search method using improved jumping frog algorithm. Eastern-European Journal of Enterprise Technologies, 4 (3 (124)), 45–53. https://doi.org/10.15587/1729-4061.2023.285292
73. Alieinykov, I., Thamer, K. A., Zhuravskyi, Y., Sova, O., Smirnova, N., Zhyvotovskyi, R. et al. (2019). Development of a method of fuzzy evaluation of information and analytical support of

strategic management. Eastern-European Journal of Enterprise Technologies, 6 (2 (102)), 16–27. https://doi.org/10.15587/1729-4061.2019.184394

74. Shyshatskyi, A. V., Zhuk, O. V., Neronov, S. M., Protas, N. M., Kashkevych, S. O. (2024). Sukupnist metodyk pidvyshchennia operatyvnosti pryiniattia rishen z vykorystanniam metaevrystychnykh alhorytmiv. Modern Aspects of Science. Scientific Perspectives, 529–557. Available at: http://perspectives.pp.ua/public/site/mono/mono-40.pdf
75. Shyshatskyi, A. V., Matsyi, O. B., Yashchenok, V. Zh., Trotsko, O. O., Kashkevych, S. O. (2024). Sukupnist metodyk pidvyshchennia operatyvnosti pryiniattia rishen z vykorystanniam kombinovanykh metaevrystychnykh alhorytmiv. Modern Aspects of Science. Scientific Perspectives, 558–594. Available at: http://perspectives.pp.ua/public/site/mono/mono-40.pdf
76. Kashkevich, S., Dmytriieva, O., Trotsko, O., Shknai, O., Shyshatskyi, A. (2023). Mathematical model of information conflict of information networks. ScienceRise, 1, 3–13. https://doi.org/10.21303/2313-8416.2024.003395
77. Kashkevych, S. O., Dmytriieva, O. I., Yefymenko, O. V., Pliekhova, H. A., Shyshatskyi, A. V. (2024). Metody otsinky stanu skladnykh dynamichnykh obiektivz vykorystanniam bioinspirovanykh alhorytmiv. Modern Aspects of Science. Scientific Perspectives, 138–177. Available at: http://perspectives.pp.ua/public/site/mono/mono-40.pdf
78. Kashkevych, S. O., Dmytriieva, O. I., Trotsko, O. O., Shknai, O. V., Shyshatskyi, A. V. (2024). Metod samoorhanizatsii informatsiinykh merezh v umovakh destabilizuiuchykh vplyviv. The development of technical, agricultural and applied sciences as the main factor in improving life. Boston: Primedia eLaunch, 192–218. https://doi.org/10.46299/ISG.2024.MONO.TECH.2.7.2
79. Shyshatskyi, A., Dmytriieva, O., Lytvynenko, O., Borysov, I., Vakulenko, Y., Mukashev, T. et al. (2024). Development of a method for assessing the state of dynamic objects using a combined swarm algorithm. Eastern-European Journal of Enterprise Technologies, 3 (4 (129)), 44–54. https://doi.org/10.15587/1729-4061.2024.304131
80. Kashkevich, S., Litvinenko, O., Shyshatskyi, A., Salnyk, S., Velychko, V. (2024). The method of self-organization of information networks in the conditions of the complex influence of destabilizing factors. Advanced Information Systems, 8 (3), 59–71. https://doi.org/10.20998/2522-9052.2024.3.07
81. Kashkevych, S. O., Dmytriieva, O. I., Plekhova, H. A., Protas, N. M., Neronov, S. M., Shyshatskyi, A. V. (2024). Naukovo-metodychnyi pidkhid z pidvyshchennia operatyvnosti obrobky riznotypnykh danykh z vykorystanniam metaevrystychnykh alhorytmiv. Modern Aspects of Science. Scientific Perspectives, 510–543. Available at: http://perspectives.pp.ua/public/site/mono/mono-40.pdf
82. Litvinenko, O., Kashkevich, S., Dmytriieva, O., Neronov, S., Plekhova, G.; Shyshatskyi, A. (Ed.) (2024). The method of self-organization of information networks in the conditions of destabilizing influences. Information and control systems: modelling and optimizations. Kharkiv: TECHNOLOGY CENTER PC, 3–34. https://doi.org/10.15587/978-617-8360-04-7.ch1
83. Kashkevich, S., Dmytriieva, O., Neronov, S., Plekhova, G., Zhyvylo, Y.; Shyshatskyi, A. (Ed.) (2024). The development of management methods based on bio-inspired algorithms.

CHAPTER 3

Information and control systems: modelling and optimizations. Kharkiv: TECHNOLOGY CENTER PC, 35–69. https://doi.org/10.15587/978-617-8360-04-7.ch2
84. Kashkevich, S., Plekhova, G., Kuchuk, N., Kuvshynov, O., Veretnov, A., Yefymenko, O.; Shyshatskyi, A. (Ed.) (2024). The development of methods for evaluating the state of complex technical systems using artificial intelligence theory. Information and control systems: modelling and optimizations. Kharkiv: TECHNOLOGY CENTER PC, 70–101. https://doi.org/10.15587/978-617-8360-04-7.ch3
85. Kashkevich, S., Plekhova, G., Yefymenko, O., Kuchuk, H., Davydov, V., Beketov, Y.; Shyshatskyi, A. (Ed.) (2024). The development of methods of learning artificial neural networks of intelligent decision-making support systems. Information and control systems: modelling and optimizations. Kharkiv: TECHNOLOGY CENTER PC, 102–137. https://doi.org/10.15587/978-617-8360-04-7.ch4
86. Kashkevich, S., Dmytriiev, I., Shevchenko, I., Lytvynenko, O., Shabanova-Kushnarenko, L., Apenko, N; Shyshatskyi, A. (Ed.) (2024). Scientific-method apparatus for improving the efficiency of information processing using artificial intelligence. Information and control systems: modelling and optimizations. Kharkiv: TECHNOLOGY CENTER PC, 138–167. https://doi.org/10.15587/978-617-8360-04-7.ch5