DOI: 10.15587/978-617-8360-23-8.CH1

1 SCIENTIFIC AND METHODOLOGICAL FRAMEWORK FOR PROCESSING HETEROGENEOUS DATA IN DECISION SUPPORT SYSTEMS

Yurii Zhuravskyi, Dmytro Fedorchuk, Oleksandr Perehuda, Mykola Romanchuk, Roman Stavisiuk, Dmitro Stupak

ABSTRACT

This section of the study presents a scientific and methodological framework for processing heterogeneous data within decision support systems. The research is grounded in the theory of artificial intelligence, specifically focusing on evolving artificial neural networks, fundamental procedures of genetic algorithms, as well as advanced hybrid bio-inspired algorithms.

In the course of the study, the authors propose the following:

– a method for processing heterogeneous data in organizational and technical systems;

– a method for evaluating the reliability of special-purpose radio communication systems using artificial intelligence theory.

The implementation of the proposed scientific and methodological framework enables the following:

– reduction of the probability of premature convergence of the metaheuristic algorithm within decision support systems;

– maintenance of a balance between convergence speed and diversity of the metaheuristic algorithm during decision-making processes;

– consideration of the type of uncertainty and data noise in the metaheuristic algorithm when operating within decision support systems;

– accounting for available computational resources of the decision support system;

– prioritization of search processes by agents within the swarms of the metaheuristic algorithm;

– initialization of swarm individuals with consideration of the type of uncertainty present in the system;

– precise training of individuals in metaheuristic algorithms;

– execution of both local and global searches considering the level of noise in the data describing the analyzed object;

– application as a universal tool for assessing the state of analysis objects through hierarchical object representation;

– verification of the reliability of the obtained results;

– enhancement of the reliability of object state assessments by constructing object-oriented and relational models of the object's state with varying levels of hierarchy;

– avoidance of the local optimum problem.

KEYWORDS

Bio-inspired algorithms, multi-agent systems, hybrid systems, reliability, efficiency, decision support systems.

The problem of enhancing the efficiency of heterogeneous data processing in decision support systems (DSS) has become increasingly urgent in modern information and automated systems of various functional purposes. The experience of recent conflicts involving the use of advanced information and automated systems demonstrates that existing methods of processing heterogeneous data allow for the processing of only 5 to 10% of the data circulating within these systems [1–5]. This limitation is attributed to several factors:

– the significant role of the human factor in the processing of heterogeneous data circulating in information and automated systems [6–10];

– the large number of heterogeneous information sources integrated into information and automated systems [11–16];

– the processing of heterogeneous data under conditions of uncertainty, which intr– oduces delays in data handling [17–21];

the presence of a substantial amount of destabilizing data that adversely affects the speed of heterogeneous data processing [22–27];

– the coexistence of structured and unstructured data within these systems, both of which require processing [29–33].

Given the diversity, the abundance of destabilizing factors, and the varying dimensionality of the indicators describing them, the need to process large volumes of heterogeneous data necessitates the development of novel approaches. One such approach is the use of metaheuristic algorithms [34–38]. While the use of canonical metaheuristic algorithms improves the efficiency of heterogeneous data processing, further improvements are limited if only their standard forms are employed [39–44]. This necessitates the introduction of diverse strategies to enhance the convergence speed and accuracy of core metaheuristic algorithms when processing heterogeneous data [44–49]. One way to improve the efficiency of heterogeneous data processing using metaheuristic algorithms is through their further refinement—by combining, comparing, and developing new procedures for their integrated application [50–54]. An analysis of previous studies [1–71] reveals several common shortcomings:

– lack of capability to construct a hierarchical system of indicators for evaluating heterogeneous data processing;

– lack of consideration for the computational resources of the system managing the data processing;

– absence of mechanisms to adjust the set of indicators governing the data processing;

– lack of mechanisms for deep learning of knowledge bases;

– high computational complexity;

– insufficient consideration of available computational (hardware) resources in DSS;

– absence of search prioritization mechanisms in specific directions.

1.1 DEVELOPMENT OF A METHOD FOR PROCESSING HETEROGENEOUS DATA IN ORGANIZATIONAL AND TECHNICAL SYSTEMS

The objective of this study is to develop a method for processing heterogeneous data in organizational and technical systems. This will enhance the efficiency of data processing within such systems while ensuring a predetermined level of reliability and enabling subsequent managerial decisions based on the processed heterogeneous data. It will also serve as a foundation for the development (or improvement) of software tools tailored for heterogeneous data processing in organizational and technical systems.

To achieve this objective, the following tasks were defined:

– to determine the implementation algorithm of the proposed method;

– to present a practical example of applying the method to process heterogeneous data in organizational and technical systems.

The object of the study is heterogeneous data within organizational and technical systems.

The problem addressed in the research is increasing the speed of processing heterogeneous data in organizational and technical systems while maintaining a specified level of reliability, regardless of the data volume received at the system's input.

The subject of the research is the process of processing heterogeneous data, which involves:

– an improved hybrid algorithm that increases processing efficiency through a competition strategy among individuals within the hybrid algorithm;

– evolving artificial neural networks used for deep learning of the knowledge base in a multi-agent system, enabling the training of both parameters and architecture of the artificial neural networks.

The hypothesis of the study posits that it is possible to increase the speed of decision-making during the processing of heterogeneous data – while maintaining a specified reliability level – through the application of an improved hybrid algorithm.

The proposed method was modeled in the Microsoft Visual Studio 2022 (USA) software environment. The simulation task focused on determining the composition of a military (force) grouping. The hardware used for the research was based on the AMD Ryzen 5 processor.

Parameters of the improved algorithm:

1. Number of iterations: 50.

2. Number of individuals in the swarm: 25.

3. Feature space range: [–150, 150].

The method for processing heterogeneous data in organizational and technical systems includes the following sequence of steps:

Step 1. Input of initial data.

At this stage, the initial data regarding the organizational and technical system are entered, including: the number and types of components within the system, the type of data circulating through the system, available computational resources, the quantity and nature of interconnections between system elements, technical specifications of control and data transmission channels, environmental application parameters, and more.

Step 2. Initialization and formation of each agent group within the hybrid algorithm.

At this stage, initial random solution sets are generated to represent agent groups of the hybrid algorithm.

The mathematical representation of a randomly selected group of agents from the set of possible agents within a defined area is described as follows

$$P_{i,j} = P_{i,j}^{\min} + \left(\lambda\left(P_{i,j}^{\max} - P_{i,j}^{\min}\right)\right)\gamma, \tag{1.1}$$

where λ – a random number within the interval [0, 1], $P_{i,j}$ – i-th element of the j-th agent group within the hybrid algorithm. The population of hybrid algorithm agents is arranged in ascending order based on $f\left(P_i\right)$, the best and worst solutions are selected $\left(P_i^{best}\right)$ and $\left(P_i^{worst}\right)$, γ – represents the degree of uncertainty in the data circulating within the organizational and technical system. At this stage, the objective function for heterogeneous data processing *f(P)* is defined, along with the population size *m* of the hybrid algorithm's swarm, the number of variables *n*, bounds for the variables *(LB, UB)*, and the termination criterion (FE_{max}') of the algorithm.

Step 3. Numbering of hybrid algorithm agents within the population, i, where $i \in [0, S]$.

Each agent in the hybrid algorithm's swarm is assigned a sequential identifier.

Step 4. Initialization of the agents' initial velocities.

The initial velocity v_0 for each agent in the population is computed using the following expression

$$v_i = \left(v_1, v_2 \ldots v_S\right),\ v_i = v_0. \tag{1.2}$$

Step 5. Preliminary assessment of the search area by the hybrid algorithm's swarm agents.

In this step, the search area is defined in natural language as the aura surrounding each group of the hybrid algorithm's swarm.

Step 6. Classification of food sources for the swarm agents.

The best food source (i.e., the one with minimal fitness value) is labeled as (FS_{ht}), representing nearby and energy-efficient resources. High-cost, desirable food sources are labeled as $FS_{at,}$ while survival-level, non-priority resources are marked as $FS_{nt.}$:

$$FS_{ht} = FS\left(sorte\_index\left(1\right)\right), \tag{1.3}$$

$$FS_{at}\left(1:4\right) = FS\left(sorte\_index\left(1:3\right)\right), \tag{1.4}$$

$$FS_{nt}\left(1:NP-4\right) = FS\left(sorte\_index\left(6:NP\right)\right). \tag{1.5}$$

Step 7. Execution of the cheetah swarm algorithm procedures.

Step 7.1 Search behavior of cheetah agents.

This behavior simulates the process of prey scouting. Cheetahs apply two main strategies: waiting and active exploration. In this context, the search space represents the problem-solving space for heterogeneous data processing. The behavior is modeled by

$$X_{i,j}^{t+1} = X_{i,j}^{t} + \hat{r}_{i,j}^{-1}\alpha_{i,j}^{t}, \tag{1.6}$$

where $X_{i,j}^{t}$ – denotes the current position of agent *i* or individual cheetah agents *i* (*i*=1, 2..., *n*) in population *j* or in the search space *j* (*j*=1, 2..., *d*), where *n* – the number of cheetah agents in the population, and d is the dimensionality of the optimization problem. $X_{i,j}^{t+1}$ – the next position of the cheetah agent, t – the cheetah agents' hunting time, T – the maximum hunting time, $\hat{r}_{i,j}^{-1}$ – randomization parameter, $\alpha_{i,j}^{t}$ – the step size for cheetah agent i in population j. The randomization parameter $\hat{r}_{i,j}$ follows a standard normal distribution. The step size $\alpha_{i,j}^{t} > 0$ is set to 0.001 x t/T, as cheetahs are slow searchers.

Step 7.2. Waiting strategy of cheetah agents.

The waiting strategy is formally defined as follows

$$X_{i,j}^{t+1} = X_{i,j}^{t}, \tag{1.7}$$

where $X_{i,j}^{t+1}$ $X_{i,j}^{t}$ – represents the new position of the cheetah agent i in population j, respectively. The proposed strategy introduces a specific coordination approach in the optimization algorithm, where all agents in the group attack simultaneously. This significantly increases the probability of successful hunting and reduces the risk of premature convergence toward food sources.

Step 7.3. Active attack behavior of cheetah agents.

This procedure outlines two key characteristics of the active hunting stage: speed and flexibility. It is important to note that cheetahs do not use group tactics while hunting, and all their attack strategies can be mathematically represented as

$$X_{i,j}^{t+1} = X_{B,j}^{t} + \hat{r}_{i,j} \cdot \beta_{i,j}^{t}, \tag{1.8}$$

where $X_{B,j}^{t}$ – the current prey position in search space j; $\hat{r}_{i,j}$, $\beta_{i,j}^{t}$ – a rotation and interaction factors associated with cheetah i in search space j, $\hat{r}_{i,j}$ – a random value described by $\left|\hat{r}_{i,j}\right|^{\exp\left(r_{i,j}/2\right)} \sin\left(2\pi r_{i,j}\right)$ and $r_{i,j}$ – a random number from the normal distribution.

Step 8. Execution of particle swarm algorithm procedures.

8.1. Velocity update of particle swarm agents.

The velocity generation of particle swarm agents is calculated based on two parameters: the global best particle G_{best} and the local best particle L_{best}. Velocities are updated using the following equation

$$V_i^{t+1} = \omega V_i^t + c_1 r_{1i}^t \left(L_{best}^t - x_i^t\right) + c_2 r_{2i}^t \left(G_{best}^t - x_i^t\right), \tag{1.9}$$

where V_i^{t+1} – the particle's velocity at iteration (t+1), V_i^t – the velocity at the previous iteration, x_i^t, r_{1i}^t, r_{2i}^t – d-dimensional vectors of uniformly distributed random numbers between 0 and 1, representing the position of particle i; c_1 and c_2 – learning coefficients, and ω – the inertia weight, typically set to 1.

8.2. Particle movement in the search space.

The new position of the particles is generated according to equation (1.10), where x_i^{t+1} – the new particle position, x_i^t – the previous position, and V_i^{t+1} – the velocity calculated using equation (1.9)

$$x_i^{t+1} = x_i^t + V_i^{t+1}. \tag{1.10}$$

Step 9. Integration of search strategies from both algorithms.

After generating the initial population, each agent receives a subpopulation size equal to half of the original population, as defined in this study. The application process of metaheuristic operators is simplified by the sequential execution of cheetah swarm behavior and particle swarm algorithm procedures. The integration procedure of both behavioral strategies is modified as follows

$$x_{i+1}^{k}=x_{i}^{k}\alpha+1-\alpha x_{best}^{k}M_{i}^{k}. \tag{1.11}$$

where x_{i+1}^{k} – the new candidate solution position x_{i}^{k}. The scaling factor α set to 0.1 in this study, x_{best}^{k} – the best solution at iteration k; M_{i}^{k} – the modulation variable of the candidate from the agent swarm. Equation (1.8) defines the combined swarm population of the hybrid algorithm, which exhibits the best performance.

Step 9.1. Modulation of metaheuristic operators.

In this study, the influence modulation of each metaheuristic operator is determined not only by comparing it to the best candidate solution but also by analyzing its elite behavior. The competition begins by identifying the solution x_{c}^{k} based on the actual solution obtained x_{i}^{k}. The solution x_{c}^{k} must only satisfy one condition: x_{c}^{k} it must be different from x_{i}^{k}.

Step 9.2. Pairwise competition of agent groups in the hybrid algorithm.

The group competition procedure in the hybrid algorithm is defined by equation (1.12):

$$\begin{aligned}&\text{if } f\left(x_{i}^{k}\right)<f\left(x_{c}^{k}\right) \text{ then } x_{i}^{k}=x_{c}^{k} \text{ and } M_{i}^{k};\\&\text{if } \left(x_{i}^{k}\right)>f\left(x_{c}^{k}\right) \text{and } Pr>r \text{ then } x_{i}^{k}=G\left(x_{c}^{k}\right) \text{ and } M_{i}^{k}=M_{c}^{k}.\end{aligned} \tag{1.12}$$

Additionally, the probabilistic threshold is determined by the performance difference between the obtained solution and the best-known solution, which varies during iterations. This threshold is computed as

$$Pr=\left|\frac{f\left(x_{i}^{k}\right)-f\left(x_{c}^{k}\right)}{BF}\right|, \tag{1.13}$$

where Pr – the probabilistic threshold, x_{i}^{k} – the actual solution, x_{c}^{k} – the reference solution, and BF – the objective value of the obtained solution. The new position x_{i}^{k} is determined based on the Euclidean distance between x_{i}^{k} and x_{c}^{k} and is updated using equation (1.14)

$$r\cdot dist-x_{c}^{k}, \tag{1.14}$$

where r – a normally distributed random number, and $dist$ – the Euclidean distance between x_{i}^{k} and x_{c}^{k}. It is worth noting that the described procedure facilitates the exploration of new regions within the solution search space x_{i}^{k}. This approach prevents premature convergence and ensures a thorough evaluation of the algorithm's computational capabilities.

Step 10. Termination condition check for hybrid swarm agents.

The algorithm terminates when the maximum number of iterations is reached. Otherwise, new candidate positions are generated, and the process is repeated.

Step 11. Knowledge base learning for hybrid swarm agents.

In the proposed study, the knowledge base of each agent in the hybrid swarm algorithm is trained using the evolving artificial neural network method described in [2]. This method modifies the movement behavior of each agent in the hybrid algorithm, contributing to more accurate analytical outcomes.

Step 12. Determination of required computational resources for the intelligent decision support system.

To prevent computational loops through *Steps 1–11* and to improve computational efficiency, the system workload is monitored. If a defined computational complexity threshold is exceeded, the number of additional hardware and software resources required is determined using the method proposed in [23].

End of Algorithm.

Proposed method for processing heterogeneous data in organizational-technical systems.

The efficiency of the proposed method for processing heterogeneous data in organizational-technical systems is compared using a set of benchmark functions, the structure of which is presented in **Table 1.1**.

Table 1.1 Evaluation of the efficiency of the proposed method for processing heterogeneous data according to the criterion of information processing speed

Function name	Metric	Canonical particle swarm algorithm	Ant colony algorithm	Black widow algorithm	Gray wolf pack algorithm	Cheetah pack algorithm	Proposed method
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
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

Continuation of Table 1.1

1	2	3	4	5	6	7	8
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 1.2 presents the results of the reliability assessment of decisions made by each of the optimization methods for processing heterogeneous data in organizational-technical systems.

Table 1.2 Evaluation of the proposed management method's efficiency based on the information processing reliability criterion

Function name	Metric	Canonical particle swarm algorithm	Ant colony algorithm	Black widow algorithm	Gray wolf pack algorithm	Cheetah pack algorithm	Proposed method
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

Continuation of Table 1.2

1	2	3	4	5	6	7	8
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
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

An analysis of **Tables 1.1** and **1.2** shows that the proposed method ensures stable algorithm performance for both unimodal and multimodal benchmark functions.

As illustrated in **Tables 1.1, 1.2**, the improvement in decision-making speed reaches 14–20% due to the application of additional procedures and the reliability of the obtained decisions being maintained at a level of 0.9.

The advantages of the proposed method are as follows:

– the initial population of agents in the hybrid swarm algorithm and their starting positions in the search space are determined considering the uncertainty degree of the initial data circulating in the organizational-technical system (1.1), through the application of correction coefficients. This contrasts with approaches in [9, 14, 20] and allows for reduced time in configuring the heterogeneous data processing subsystem during its initial setup;

– the initial velocity of each agent in the hybrid swarm is taken into account (1.2), enabling the prioritization of searches in specific dimensions of the search space (by elements and components of the organizational-technical system), compared to methods in [9–15];

– the suitability of the decisions made during heterogeneous data processing is evaluated considering the aggregate of external factors, thereby reducing the overall decision search time (*Step 5*), compared to [14, 16, 17];

– the search strategies of food source localization for agents in the hybrid swarm algorithm are versatile, allowing classification of the conditions and factors influencing heterogeneous data processing (*Step 6*), compared to [14, 16, 17]. This enables the identification of the most suitable decision-making options according to the defined optimization criterion;

– the method allows exploration of solution spaces defined by non-standard functions by using a step selection procedure for cheetah agents within the hybrid swarm algorithm (*Step 7*), in contrast to [9, 12–18];

– replacement of unfit agents is possible via population updating mechanisms of the hybrid swarm algorithm (*Steps 8–10*), compared to [9, 12–18];

– the method supports comparative assessment of heterogeneous data processing efficiency using the metaheuristic operator modulation procedure (*Step 9.1*), as compared to [20].

– the capability of simultaneously searching for solutions in multiple directions is supported (*Steps 1–12*, **Tables 1.1** and **1.2**).

– the method provides for deep learning of the knowledge bases of agents in the hybrid swarm algorithm (*Step 10*), compared to [9–20];

– the required number of computational resources can be estimated in cases where available computing capacity is insufficient (*Step 12*), compared to [9–20].

The drawbacks of the proposed method include:

– loss of informativeness when processing heterogeneous data due to the construction of a membership function;

– lower accuracy in evaluating individual parameters of the heterogeneous data processing state;

– decreased reliability of the obtained decisions when searching in multiple directions simultaneously;

– lower evaluation precision compared to other methods for processing heterogeneous data.

The proposed method enables:

– determination of the optimal efficiency indicator for heterogeneous data processing based on the selected optimization criterion;

– identification of effective measures to improve the efficiency of heterogeneous data processing;

– increased speed of heterogeneous data processing while maintaining the required decision-making reliability;

– reduced use of computational resources in decision support systems.

The limitations of the study include the requirement for information on the degree of uncertainty in the data circulating within organizational-technical systems, and the need to consider delays in the collection and dissemination of information from the system components.

The proposed approach is recommended for solving problems related to the processing of heterogeneous data characterized by a high degree of complexity.

1.2 METHOD FOR ASSESSING THE RELIABILITY OF SPECIAL-PURPOSE RADIO COMMUNICATION SYSTEMS USING ARTIFICIAL INTELLIGENCE THEORY

Radio communication systems currently serve as the core of the transmission environment and are used to transfer all types of traffic. The method for assessing the reliability of special-purpose radio communication systems (SP-RCS) using artificial intelligence theory consists of the following sequence of steps:

Step 1. Input of initial data.

At this stage, the available initial data concerning the SP-RCS and the enemy's electronic warfare (EW) assets are entered, specifically:

– the quantity and types of radio communication assets included in the system;
– the quantity and types of enemy EW assets;
– technical characteristics of the SP-RCS;
– technical characteristics of the EW assets;
– architecture (topology of connections) of the SP-RCS;
– architecture (topology of connections) of the EW assets;
– types of data circulating within the system;
– available computational resources;
– information about the operating environment, etc.

Step 2. Initialization and formation of the general agent population for the hybrid swarm algorithm.

At this stage, initial random sets of solutions are generated, representing groups of agents in the hybrid swarm algorithm.

The mathematical representation of a randomly selected group of agents from the hybrid algorithm, taken from the set of all possible agents within a defined area, is described as follows

$$P_{i,j} = P_{i,j}^{\min} + \left(\lambda\left(P_{i,j}^{\max} - P_{i,j}^{\min}\right)\right)\gamma, \tag{1.15}$$

where λ – a random number in the range [0, 1], $P_{i,j}$ – i-th identifier of the j-th agent group of the hybrid algorithm. The agents of the hybrid algorithm are arranged in ascending order of values $f(P_i)$, with the best $\left(P_i^{best}\right)$ and the worst solutions selected $\left(P_i^{worst}\right)$. γ represents the degree of uncertainty regarding the enemy's electronic warfare (EW) assets. At this stage, the target reliability function $f(P)$ is also defined, as well as the population size *(m)* of the hybrid swarm, the number of variables *(n)*, bounds on variable values *(LB, UB)*, and the termination criterion for the algorithm ($FE_{\max}'$).

Step 3. Numbering of hybrid algorithm agents in the population, i, $i \in [0, S]$.

Each agent in the hybrid swarm algorithm is assigned a sequential identifier within the population.

Step 4. Determination of initial agent velocity in the population.

The initial velocity v_0 of each agent in the population is calculated using the following expression

$$v_i = \left(v_1, v_2, \ldots, v_s\right), \; v_i = v_0. \tag{1.16}$$

Step 5. Preliminary evaluation of the search area by hybrid swarm agents.

In this step, the search area is linguistically defined as the *aura* surrounding each group of agents in the hybrid swarm algorithm.

Step 6. Classification of food sources for hybrid swarm agents.

The location of the best food source (i.e., the one with the lowest fitness value) is denoted as (FS_{ht}) a source that is nearby and requires minimal energy to locate and acquire. Delicacy food sources, which demand the most effort to obtain, are marked as FS_{at}.

Other non-priority food sources (required only for individual survival) are denoted as FS_{nt}, defined as follows:

$$FS_{ht} = FS\left(sorte\_index\left(1\right)\right), \tag{1.17}$$

$$FS_{at}\left(1:4\right) = FS\left(sorte\_index\left(1:3\right)\right), \tag{1.18}$$

$$FS_{nt}\left(1:NP-4\right) = FS\left(sorte\_index\left(6:NP\right)\right). \tag{1.19}$$

Step 7. Execution of calculations by individual hybrid swarm agent groups.

Step 7.1. Execution of calculations by dung beetle agents.

This algorithm mimics the natural behavior of dung beetles. The dung beetle algorithm (DBA) divides the entire population into four segments based on this behavior.

Step 7.1.1. Ball rolling procedure.

When dung beetles roll their dung balls, they ensure the path is linear, aligned with celestial cues. To simulate this behavior, agents in the algorithm move directionally through the entire search space. It is assumed that sunlight intensity affects the beetles' paths. The agent's position is updated as follows:

$$\begin{aligned} &X_n^{i+1} = X_n^i + a.k.X_n^{i-1} + b.\Delta x; \\ &\Delta x = \left|X_n^i - X^{\omega}\right|, \end{aligned} \tag{1.20}$$

where X_n^{i+1} – the position of the n-th dung beetle at the i-th iteration, $k \in \left(0, 0.2\right]$ – the deviation coefficient (assigned a value of 0.1 in the code), $b \in \left(0,1\right)$ – a natural coefficient (assigned a value of 0.3 in the code), x – the change in illumination intensity, and X^{ω} – the worst agent position in the current population. a – a natural coefficient assigned a value of either 1 or –1, where a = 1 indicates no effect from environmental interference on the beetle's direction, and a = –1 indicates deviation. In this study, a characterization of the level of SP-RCS suppression by EW assets.

In nature, when dung beetles encounter an obstacle, they rotate their bodies to alter direction and bypass the barrier. In our case, this simulates the reconfiguration of the information transmission route when an SP-RCS element is disabled by disruptive factors.

This is described by:

$$X_n^{i+1} = X_n^i + \tan\alpha \left| X_n^i - X_n^{i-1} \right|;$$
$$0 \le \alpha \le \pi, \tag{1.21}$$

where α – the deflection angle between the new direction of the beetle and its original path.

Step 7.1.2. Beetle reproduction procedure.

When the dung ball returns to the nest, the beetles must choose a suitable location for egg-laying to ensure a safe environment for their offspring. Based on the discussion above, the algorithm models this behavior by identifying a boundary region where females deposit eggs. This is defined as follows:

$$LB_1 = max\left(X^t.(1-T), LB\right),$$
$$UB_1 = min\left(X^t.(1+T), UB\right), \tag{1.22}$$

where X^t – the current local optimum, LB_1 and UB_1 – the lower and upper boundaries of the egg-laying region; Lb and Ub – the lower and upper bounds of the overall search space, respectively; and the inertia weight is $R = 1 - t/T_{max}$, where T_{max} – the maximum number of iterations during the algorithm's runtime.

In the context of this study, the procedure identifies the SP-RCS elements that are least suppressed.

The boundary range of the egg-laying region dynamically changes to prevent the algorithm from falling into a local optimum. Thus, during the iteration process, the positions of the laid dung balls can also shift. This process is described as

$$X_n^{i+1} = X^t + B_1.\left(X_n^i - LB_1\right) + B_2.\left(X_n^i - UB_1\right), \tag{1.23}$$

where X_n^i – the position of the n-th dung ball to be laid at iteration t, B_1 and B_2 – two independent random matrices, and D is the algorithm's dimensionality.

Step 7.1.3. Hatching of dung beetles.

Once the young dung beetles successfully hatch, they move out in groups to search for food. Their food-searching behavior is constrained by a limited range. For these young beetles, the optimal foraging area is determined as:

$$LB_2 = max\left(X^{tb}.(1-T), LB\right),$$
$$UB_2 = min\left(X^{tb}.(1-T), UB\right), \tag{1.24}$$

where X^{tb} – the global optimum, and LB_2 and UB_2 – the lower and upper bounds of the region, respectively. Once the location for the young dung beetle is determined, its position can be updated as follows

$$X_i^{n+1} = X_i^n + C_1.\left(X_i^n - LB_2\right) + C_2.\left(X_i^n - UB_2\right), \tag{1.25}$$

where X_i^{n+1} – represents the location information of the i-th young beetle at iteration t, C_1 is a Gaussian-distributed random number, and C_2 is a value in the interval (0, 1).

Step 7.1.4. Execution of the stealing strategy.

In dung beetle populations, some individuals steal dung balls from others. These beetles are referred to as "thieves". As mentioned earlier, X^s is the globally optimal location – that is, the best location for accessing food.

Therefore, it is assumed that the surroundings of X represent the most competitive region for food. The position of the dung beetle thieves is updated during the iteration process as follows

$$X_n^{i+1} = X^s + P.d.\left(\left|X_n^{\ i} - X^t\right| + \left|X_n^{\ i} - X^-\right|\right), \tag{1.26}$$

where d – a random vector of size $1.D$, following a normal distribution, and P – a constant.

Step 7.2. Execution of calculations by osprey swarm agents.

Step 7.2.1. Global exploitation.

To model the first stage of population updating for osprey agents, the natural behavior of ospreys was simulated. Equation (1.27) is used to determine the location of each osprey

$$OS_n = \{X_t \mid t \in \{1, 2, ..., N\} \wedge O_t < O_n\} \cup \{X^*\}, \tag{1.27}$$

where OS_n – the set of positions occupied by the n-th osprey, and X^* – the precise location of the optimal osprey. The osprey independently identifies the location of a fish and initiates its attack. The new position of the osprey relative to the fish is calculated based on modeled movement behavior, as described in equation (1.28). If the fitness function at the new position yields a better value, the previous position is replaced:

$$\begin{gathered} x_{i,j}^{P1} = xi,j + ri,j.\left(SF_{i,j} - I_{i,j}.x_{i,j}\right), \\ x_{i,j}^{P1} = \begin{cases} x_{i,j}^{P1}, lb_j \le x_{i,j}^{P1} \le ub_j; \\ lb_j, x_{i,j}^{P1} \langle lb_j; \\ ub_j, x_{i,j}^{P1} \rangle ub_j. \end{cases} \\ X_i = \begin{cases} X_i^{P1}, F_i^{P1} \langle F_i; \\ X_i, \text{also}, \end{cases} \end{gathered} \tag{1.28}$$

where $x_{i,j}^{P1}$ – the new position in the j-th dimension for the i-th osprey at stage one; $F_{i,j}$ – the fitness value of the current osprey position; $SF_{i,j}$ – a random number in the range [0, 1]; and $I_{i,j}$ – a random integer from the set 1, 2.

Step 7.2.2. Local exploitation.

When an osprey catches a fish, it transports it to a safe zone to consume it. The second stage of population updating uses simulation-based modeling to replicate the osprey's natural behavior. Guiding the fish to a suitable location leads to minor adjustments in the osprey's position within the search space.

This enhances the local search capability of the procedure and allows the algorithm to converge toward a more optimal solution in the vicinity of a previously defined result. This position is considered "suitable for fish consumption" and is determined using the following equation (1.29). Subsequently, if the fitness function value improves at this new position, the previous position of the corresponding offspring is replaced:

$$x_{i,j}^{P2} = x_{i,j} + \frac{lb_j + r \cdot \left(ub_j - lb_j\right)}{t},$$

$$i = 1,2,\ldots,N, j = 1,2,\ldots,m, t = 1,2,\ldots,T,$$

$$x_{i,j}^{P2} = \begin{cases} x_{i,j}^{P2}, lb_j \le x_{i,j}^{P2} \le ub_j; \\ lb_j, x_{i,j}^{P2} \langle lb_j; \\ ub_j, x_{i,j}^{P2} \rangle ub_j. \end{cases}$$

$$X_i = \begin{cases} X_i^{P2}, F_i^{P2} \langle F_i; \\ X_i, \text{also}, \end{cases} \tag{1.29}$$

where $x_{i,j}^{P2}$ — the new position of the *j*-th dimension for the *i*-th osprey during the second stage; $F_{i,j}^{P2}$ — the corresponding fitness value of the updated position; r is a random number in the range [0, 1]; t and T are the current and maximum number of iterations, respectively.

Step 8. Integration of the search strategies from both algorithms.

After generating the initial population, each agent receives a population size equal to half of the original population, as defined in the referenced study. The process of applying metaheuristic operators is simplified by sequentially executing the behavior of the osprey swarm algorithm and the dung beetle swarm algorithm, according to their respective procedures. The procedure for integrating both behavioral strategies is modified as follows

$$x_{i+1}^k = x_i^k \alpha + 1 - \alpha x_{best}^k M_i^k, \tag{1.30}$$

where x_{i+1}^k — the new candidate solution position x_i^k. The scaling coefficient α in this study is set to 0.1; x_{best}^k — the best solution at iteration k; M_i^k — the modulation variable of the candidate from the swarm. Equation (1.30) defines the merged population of agents in the hybrid algorithm that exhibits optimal performance.

Step 8.1. Modulation of metaheuristic operators.

In this study, the influence of each metaheuristic operator is modulated not only by the traditional comparison to the best candidate solution but also by analyzing its elite behavior. The competition begins by identifying solution x_c^k in comparison to the actual obtained solution x_i^k. The only requirement x_c^k is that it must differ from x_i^k.

Step 8.2. Pairwise competition of agent groups in the hybrid algorithm.

The group competition procedure in the hybrid algorithm is defined by Equation (1.31):

$$\text{if } f\left(x_i^k\right) < f\left(x_c^k\right) \text{ then } x_i^k = x_c^k \text{ and } M_i^k;$$
$$\text{if } \left(x_i^k\right) > f\left(x_c^k\right) \text{and } Pr > r \text{ then } x_i^k = G\left(x_c^k\right) \text{ and } M_i^k = M_c^k. \tag{1.31}$$

Additionally, the probabilistic threshold is defined as the performance difference between the obtained solution and the best solution, and it varies across iterations.

The threshold is calculated as follows

$$Pr = \left|\frac{f\left(x_i^k\right) - f\left(x_c^k\right)}{BF}\right|, \tag{1.32}$$

where Pr – the probabilistic threshold, x_i^k – the actual solution, x_c^k – the benchmark solution, and BF – the cost (fitness) of the obtained solution. The new position x_i^k is determined by calculating the Euclidean distance between x_i^k and x_c^k. A position is updated using the following formula

$$r \cdot dist - x_c^k, \tag{1.33}$$

where r – a randomly distributed number and *dist* is the Euclidean distance between x_i^k and x_c^k. It is worth noting that this procedure facilitates the exploration of new regions within the solution search space x_i^k. It prevents premature convergence and ensures a more comprehensive analysis of the algorithm's computational capabilities.

Step 9. Stopping criterion check for the hybrid swarm agents.

The algorithm terminates when the maximum number of iterations is reached. Otherwise, new positions are generated, and the condition check is repeated.

Step 10. Learning of knowledge bases for hybrid swarm agents.

In this study, the knowledge base of each agent in the hybrid swarm algorithm is trained using the evolving artificial neural network method developed in [2]. This method adjusts the movement patterns of each agent in the hybrid swarm, contributing to more accurate analytical results in future iterations.

Step 11. Determination of required computational resources for the intelligent decision support system.

To avoid computational loops through *Steps 1–10* and to increase computational efficiency, system load is monitored. If the defined computational complexity threshold is exceeded, the required number of additional software and hardware resources is determined using the method proposed in [23].

The efficiency of the proposed method for assessing the reliability of special-purpose radio communication systems using artificial intelligence theory is evaluated using a set of benchmark functions, as presented in **Table 1.3**.

Table 1.3 Evaluation of the efficiency of the proposed method for assessing the reliability of special-purpose radio communication systems using artificial intelligence theory

Function name	Metric	Canonical particle swarm algorithm	Ant colony algorithm	Black widow algorithm	Gray wolf pack algorithm	Cheetah pack algorithm	Proposed method
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
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 1.4 presents the results of the reliability assessment of decisions obtained using each of the optimization methods for processing heterogeneous data in organizational-technical systems.

Table 1.4 Evaluation of the proposed method's efficiency based on the information processing reliability criterion

Function name	Metric	Canonical particle swarm algorithm	Ant colony algorithm	Black widow algorithm	Gray wolf pack algorithm	Cheetah pack algorithm	Proposed method
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
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 1.3** and **1.4**, it can be concluded that the proposed method ensures stable algorithm performance for key unimodal and multimodal test functions.

As evident from **Tables 1.3, 1.4**, the increase in decision-making speed reaches 16–20% due to the use of additional procedures and the achievement of decision reliability at the 0.9 level.

The advantages of the proposed method are as follows:

– the initial population of agents in the hybrid algorithm swarm and their initial positions in the search space are determined considering the degree of uncertainty in the input data circulating in the organizational and technical system (equation 1.15) through the use of appropriate correction coefficients, in comparison with studies [9, 14, 20]. This reduces the time required for the initial setup of the heterogeneous data processing subsystem;

– the initial velocity of each agent in the hybrid swarm is taken into account (equation 1.16), allowing prioritization of the search in the corresponding search space (by elements and components of the organizational and technical system), compared to studies [9–15];

– the suitability of decisions made during heterogeneous data processing is assessed by considering external factors, reducing the time needed to find a solution (*Step 5*), compared to studies [14, 16, 17];

– the universality of the food source search strategies among the agents in the hybrid algorithm swarm allows for classification of the conditions and factors influencing the heterogeneous data processing process (*Step 6*), compared to studies [14, 16, 17]. This makes it possible to determine the most suitable solutions according to the specified optimization criterion;

– the ability to explore solution spaces defined by atypical functions due to the use of the cheetah agents' step-size selection procedure in the hybrid swarm (*Step 7*), compared to studies [9, 12–18];

– the replacement of ineffective individuals is carried out by updating the population of agents in the hybrid algorithm swarm (*Steps 8–10*), compared to studies [9, 12–18];

– the ability to conduct comparative evaluation of the effectiveness of heterogeneous data processing using the procedure for modulating metaheuristic operators (*Step 9.1*), compared to study [20];

– the ability to search for a solution in multiple directions simultaneously (*Steps 1–12*, **Tables 1.3** and **1.4**);

– the ability for deep learning of the knowledge base of agents in the hybrid algorithm swarm (*Step 10*), compared to studies [9–20];

– the ability to calculate the required amount of computational resources that need to be involved if it is impossible to perform calculations with the available computational resources (*Step 12*), compared to studies [9–20].

Disadvantages of the proposed method include:

– loss of informativeness during heterogeneous data processing due to the construction of a membership function;

– lower accuracy in evaluating individual parameters of the heterogeneous data processing state;

– loss of decision reliability when searching in multiple directions simultaneously;

– lower evaluation accuracy compared to other heterogeneous data processing methods.

The proposed method allows:

– determining the optimal performance indicator for heterogeneous data processing according to a defined optimization criterion;

– identifying effective measures to increase the efficiency of heterogeneous data processing;

– increasing the processing speed of heterogeneous data while ensuring the specified decision-making reliability;

– reducing the use of computational resources in decision support systems.

Limitations of the study include the need for information on the degree of uncertainty in the data circulating in organizational and technical systems, and the need to consider delays in data collection and delivery from components of these systems.

CONCLUSIONS

The algorithm for implementing a method for processing heterogeneous data in organizational and technical systems has been developed.

Through the introduction of additional and improved procedures, the following capabilities have been achieved:

– initialization of the initial population of agents in the hybrid algorithm swarm and their positioning within the search space, accounting for the uncertainty level of the input data circulating within the organizational and technical system, made possible through the application of correction coefficients. This significantly reduces the time required for the initial setup of the data processing subsystem;

– consideration of the initial velocity of each agent in the hybrid algorithm swarm, enabling prioritization of the search process across the system's components and elements;

– assessment of the suitability of the decisions made during data processing, taking into account the aggregate influence of external factors, which reduces the time required to find a solution;

– classification of the conditions and factors affecting the heterogeneous data processing process through the universality of the food-source search strategies of the hybrid swarm agents. This allows for the selection of the most suitable processing solutions according to the defined optimization criterion;

– exploration of complex solution spaces represented by non-standard functions, supported by the dynamic step-size adjustment procedure for cheetah agents within the hybrid algorithm swarm;

– replacement of unfit individuals by updating the hybrid algorithm swarm's population;

– comparative evaluation of the efficiency of heterogeneous data processing through metaheuristic operator modulation;

– capability to search for solutions in multiple directions simultaneously;

– capability for deep learning of the hybrid algorithm swarm agents' knowledge base;

– capability to estimate the required amount of computational resources necessary in cases where the existing computational capacity is insufficient.

A case study demonstrating the application of the proposed method for processing heterogeneous data in a military (forces) operational grouping confirmed an improvement in decision-making efficiency by approximately 14–20%, achieved through the integration of additional procedures and maintenance of decision reliability at a level of 0.9.

The algorithm for implementing a method for assessing the reliability of special-purpose radio communication systems using artificial intelligence theory has been developed. With the inclusion of additional and enhanced procedures, the following improvements have been achieved:

– initialization of the initial agent population and positioning within the search space, taking into account the uncertainty in the input data related to the operational environment of the radio communication systems via appropriate correction coefficients, reducing setup time;

– incorporation of each agent's initial velocity, which allows for prioritization in the respective search space;

– evaluation of decision suitability during data processing, accounting for multiple external influences, thereby accelerating solution discovery;

– classification of conditions and influencing factors on data processing based on the hybrid swarm agents' search strategy universality, improving solution suitability according to the defined optimization criterion;

– study of solution spaces described by non-standard functions via the cheetah agents' step-size adjustment mechanism;

– population update procedures that replace ineffective individuals in the swarm;

– performance comparison using metaheuristic operator modulation techniques;

– capability for multi-directional solution search;

– capability for deep training of the agents' knowledge bases;

– ability to calculate the required amount of computational resources needed when current resources are insufficient.

Another case study applying the proposed method to heterogeneous data processing in a military (forces) operational grouping demonstrated a 16–20% increase in decision-making efficiency, with a maintained decision reliability level of 0.9, attributable to the implementation of the described enhancements.

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