Edited by
Svitlana Kashkevich

DECISION SUPPORT SYSTEMS: MATHEMATICAL SUPPORT

Collective monograph

Technology Center

2025

UDC 004.8
D29

Published in 2025
by TECHNOLOGY CENTER PC®
Shatylova dacha str., 4, Kharkiv, Ukraine, 61165

D29 Authors:
Edited by **Svitlana Kashkevich**
Svitlana Kashkevich, Iraida Stanovska, Oleh Shknai, Oleksandr Lytvynenko, Yevhen Artamonov, Andrii Veretnov, Oleksandr Gaman, Andrii Radchenko, Serhii Neronov, Zaur Musaiev, Oleg Sova, Olena Nechyporuk, Oleksii Nalapko, Oleksiy Buyalo, Anna Lyashenko, Olha Matsyi, Anastasiia Voznytsia, Daniil Krant, Kostiantin Radchenko, Olena Shaposhnikova, Nataliia Saienko, Viktoriia Shevchenko, Nataliia Holeho, Taras Zaloznyi, Oleksandr Yaryzhko, Olena Nosyryeva, Anzhela Popova, Ihor Pimonov, Dmitry Leinyk
Decision support systems: mathematical support: collective monograph. – Kharkiv: TECHNOLOGY CENTER PC, 2025. – 202 p.

This collective monograph is the result of the painstaking work of the authors of the study on solving the problems of mathematical support for decision-making using the theory of artificial intelligence. Structurally and logically, the monograph is divided into sections that are structured in a certain thematic area of research on mathematical support using artificial intelligence methods. This work is not comprehensive, and does not claim to be complete in this scientific and theoretical direction. The monograph will be useful for researchers involved in solving problems of mathematical support using the theory of artificial intelligence, the development of new (improving existing) approaches to solving complex technical problems in various fields of human activity.
The monograph is also useful for practitioners – designers, developers implementing modern solutions in the field of information technology, engaged in the development of information, information-analytical, and automated systems.
Figures 14, Tables 39, References 169 items.

DOI: 10.15587/978-617-8360-13-9
ISBN 978-617-8360-13-9 (on-line)

Cite as: Kashkevich, S. (Ed.) (2025). Decision support systems: mathematical support. Kharkiv: TECHNOLOGY CENTER PC, 202. doi: http://doi.org/10.15587/978-617-8360-13-9

9 786178 360139

AUTHORS

CHAPTER 1

SVITLANA KASHKEVICH
Senior Lecturer
Department of Intelligent Cybernetic Systems
State University "Kyiv Aviation Institute"
ORCID: https://orcid.org/0000-0002-4448-3839

IRAIDA STANOVSKA
Doctor of Technical Sciences, Professor
Department of Advanced Mathematics and Systems Modelling
Odesa Polytechnic National University
ORCID: https://orcid.org/0000-0002-5884-4228

OLEH SHKNAI
PhD, Senior Researcher, Leading Researcher
Research Institute of Military Intelligence
ORCID: https://orcid.org/0000-0002-5572-4917

OLEKSANDR LYTVYNENKO
PhD, Senior Researcher
Research Department
Research Center
Military Institute of National Taras Shevchenko University of Kyiv
ORCID: http://orcid.org/0009-0000-6541-3621

YEVHEN ARTAMONOV
PhD, Associate Professor
Department of Intelligent Cybernetic Systems
State University "Kyiv Aviation Institute"
ORCID: http://orcid.org/0000-0002-9875-7372

ANDRII VERETNOV
PhD, Leading Researcher
Research Department
Central Scientific Research Institute of Armament and Military Equipment of the Armed Forces of Ukraine
ORCID: https://orcid.org/0000-0003-0160-7325

CHAPTER 2

OLEKSANDR GAMAN
Adjunct
Scientific and Organizational Department
Kruty Heroes Military Institute of Telecommunications and Information Technology
ORCID: https://orcid.org/0000-0003-4676-3321

SVITLANA KASHKEVICH
Senior Lecturer
Department of Intelligent Cybernetic Systems
State University "Kyiv Aviation Institute"
ORCID: https://orcid.org/0000-0002-4448-3839

OLEH SHKNAI
PhD, Senior Researcher, Leading Researcher
Research Institute of Military Intelligence
ORCID: https://orcid.org/0000-0002-5572-4917

ANDRII RADCHENKO
PhD Student
Faculty of Computer Sciences and Technologies
State University "Kyiv Aviation Institute"
ORCID: https://orcid.org/0009-0003-8288-171X

SERHII NERONOV
Senior Lecturer
Computer Science and Information Systems Department
Kharkiv National Automobile and Highway University
ORCID: https://orcid.org/0000-0003-2381-1271

ZAUR MUSAIEV
PhD, Assistant
Department of Operation, Testing, Service of Construction and Road Machinery
Kharkiv National Automobile and Highway University
ORCID: https://orcid.org/0000-0002-5533-0897

CHAPTER 3

OLEH SHKNAI
PhD, Senior Researcher, Leading Researcher
Research Institute of Military Intelligence
ORCID ID: https://orcid.org/0000-0002-5572-4917

OLEG SOVA
Doctor of Technical Science, Professor, Deputy Head of Scientific Center
Scientific Center for Building Integrity and Preventing Corruption in the Security and Defense Sector
The National University of Defense of Ukraine
ORCID ID: https://orcid.org/0000-0002-7200-8955

OLENA NECHYPORUK
Doctor of Technical Sciences, Professor, Head of Department
Department of Intelligent Cybernetic Systems
State University "Kyiv Aviation Institute"
ORCID ID: https://orcid.org/0000-0001-8203-7998

OLEKSII NALAPKO
PhD, Head of Research Department
Research Department for Development of Electronic Warfare Equipment
Central Scientific Research Institute of Armament and Military Equipment of the Armed Forces of Ukraine
ORCID ID: https://orcid.org/0000-0002-3515-2026

OLEKSIY BUYALO
PhD, Senior Researcher, Senior Lecturer
Yevhenii Bereznyak Military Academy
ORCID ID: https://orcid.org/0000-0002-8848-864X

ANNA LYASHENKO
Senior Researcher
Scientific Center
Kruty Heroes Military Institute of Telecommunications and Information Technology
ORCID ID: https://orcid.org/0000-0002-5318-8663

CHAPTER 4

SVITLANA KASHKEVICH
Senior Lecturer
Department of Intelligent Cybernetic Systems
State University "Kyiv Aviation Institute"
ORCID ID: https://orcid.org/0000-0002-4448-3839

OLHA MATSYI
PhD, Associate Professor
Department of Mathematical Modeling and Data Analysis
V. N. Karazin Kharkiv National University
ORCID ID: https://orcid.org/0000-0002-1350-9418

ANASTASIIA VOZNYTSIA
PhD Student
State University "Kyiv Aviation Institute"
ORCID ID: https://orcid.org/0009-0004-3767-7354

OLEKSIY BUYALO
PhD, Senior Researcher, Senior Lecturer
Yevhenii Bereznyak Military Academy
ORCID ID https://orcid.org/0000-0002-8848-864X

DANIIL KRANT
PhD Student
State University "Kyiv Aviation Institute"
ORCID ID: https://orcid.org/0000-0003-4601-2827

KOSTIANTIN RADCHENKO
PhD Student
State Non-Commercial Company State University "Kyiv Aviation Institute"
ORCID ID: https://orcid.org/0009-0009-8056-6293

CHAPTER 5

SVITLANA KASHKEVICH
Senior Lecturer
Department of Intelligent Cybernetic Systems
State University "Kyiv Aviation Institute"
ORCID ID: https://orcid.org/0000-0002-4448-3839

OLENA SHAPOSHNIKOVA
PhD, Associate Professor
Department of Computer Science and Information Systems
Kharkiv National Automobile and Highway University
ORCID ID: https://orcid.org/0000-0002-0405-8205

NATALIIA SAIENKO
Doctor of Pedagogical Sciences, Professor
Department of Foreign Languages
Kharkiv National Automobile and Highway University
ORCID ID: https://orcid.org/0000-0001-7953-3747

VIKTORIIA SHEVCHENKO
PhD, Associate Professor
Department of Computer Science and Information Systems
Kharkiv National Automobile and Highway University
ORCID ID: https://orcid.org/0000-0003-0620-6811

NATALIIA HOLEHO
Senior Lecturer
Department of Intelligent Cybernetic Systems
State University "Kyiv Aviation Institute"
ORCID ID: https://orcid.org/0000-0002-6430-8659

TARAS ZALOZNYI
PhD Student
State University "Kyiv Aviation Institute"
ORCID ID: https://orcid.org/0009-0002-3012-4840

CHAPTER 6

SVITLANA KASHKEVICH
Senior Lecturer
Department of Intelligent Cybernetic Systems
State University "Kyiv Aviation Institute"
ORCID: https://orcid.org/0000-0002-4448-3839

OLEKSANDR YARYZHKO
PhD, Associate Professor
Department of Operation, Testing, Service of Construction and Road Machinery
Kharkiv National Automobile and Highway University
ORCID: https://orcid.org/0000-0001-6398-8472

OLENA NOSYRYEVA
Senior Lecturer
Department of Foreign Languages
Kharkiv National Automobile and Highway University
ORCID: https://orcid.org/0009-0009-5094-496X

ANZHELA POPOVA
PhD, Associate Professor
Department of Computer Science and Information Systems
Kharkiv National Automobile and Highway University
ORCID: https://orcid.org/0000-0003-4013-5244

IHOR PIMONOV
PhD, Associate Professor
Department of Operation, Testing, Service of Construction and Road Machinery
Kharkiv National Automobile and Highway University
ORCID: https://orcid.org/0000-0001-6100-3529

DMITRY LEINYK
Junior Researcher
Research Department
Research Institute of Military Intelligence
ORCID: https://orcid.org/0009-0004-5008-8837

ABSTRACT

The high dynamism of the development of social processes and phenomena determines the formation of a new system of the worldview of mankind, the modification (change) of the hierarchy of needs and values, challenges to the pace and quality of development.

Solving complex problems associated with meeting the requirements of our time requires the use of innovative scientific approaches. Today, the use of modern intellectual technologies, such as neural networks, deep learning and artificial intelligence, is a prerequisite for the proactive development of all spheres of human activity: medicine, technology, business, environmental protection, education, transport and communication, etc. Thus, the intellectualization of technical and managerial systems can be considered one of the key foundations of the new paradigm of science and technology. The phrase "artificial intelligence systems" today is understandable to everyone. The context of this term is associated with such concepts as robotics, forecasting, processing of large information flows, expert systems, diagnostics, smart home or smart tools projects, cyberphysical space and cyberphysical systems, computer translation, etc.

There is a positive dynamics of the development and implementation of artificial intelligence elements in most types of software: mobile applications, information systems, electronic devices, etc.

This process of "intellectualization" allows us to talk about a gradual increase in the intelligence of modern computer systems capable of performing functions that are traditionally considered intellectual: understanding language, logical inference, using the accumulated knowledge, learning, pattern recognition, as well as learning and explaining their decisions.

The monograph presents the scientific and methodological apparatus of intellectual assessment of the state of complex systems. During the study, the authors proposed: a methodological approach to assess the state of hierarchical systems using a metaheuristic algorithm; a technique for analyzing and predicting the state of multidimensional objects using a metaheuristic algorithm and a technique for increasing the reliability of estimating the state of an object.

A separate section presents a comprehensive model for processing diverse data in intelligent decision support systems; method of processing different types of data in intelligent network and server architecture management systems, as well as a technique for increasing the efficiency of processing different types of data in intelligent network and server architecture management systems.

The next section of this monograph proposes a set of methods for improving the efficiency of information processing in intelligent decision support systems. During the study, the authors proposed: a method for managing information flows in intelligent decision support systems using a population algorithm; method of evaluating the efficiency of processing different types of data in decision support systems; method of evaluation and forecasting in intelligent decision support systems.

A separate section of the study proposes a scientific and methodological apparatus for processing various types of data in automated control systems. feasibility of using artificial intelligence theory for processing different types of data in automated control systems is substantiated; developed a method of data distribution in automated control systems; a model of the process of evaluating the processing of different types of data in automated command and control systems using expert information has been developed, and the methodology for setting up an information system for evaluating the process of processing different types of data in automated control systems under conditions of uncertainty has been improved.

The next section of the monograph is devoted to intellectual methods for assessing the state of channels of unmanned aerial vehicles. In the course of the study, the authors proposed: the method of intellectual assessment of the state of channels of unmanned aerial vehicles (UAV), based on the use of fuzzy sets and artificial neural networks, allows, with its sufficient simplicity, to obtain sufficiently accurate solutions and methods for identifying the state of control channels and data transmission UAV.

The monograph also offers intellectual methods for assessing the state of hierarchical systems. In the course of this study, the authors: the analysis of knowledge representation models was carried out, the advantages of using the production representation of knowledge in expert systems were substantiated; method of evaluation and prediction using fuzzy cognitive maps was developed. The study developed a method of visualizing the states of a hierarchical system. A method of evaluating complex hierarchical systems based on an improved swarm of particles has been developed. The presented estimation method is based on a combination of particle swarm methods and coordinate averaging and its modification using several particle swarms and the inclusion of the Hook-Jeeves procedure and corresponding correction coefficients.

The monograph will be useful for researchers involved in solving optimization problems, using the theory of artificial intelligence, developing new (improving existing) approaches to solving complex technical problems in various fields of human activity.

The monograph is also useful for practitioners – designers, developers implementing modern solutions in the field of information technology, engaged in the development of information, information and analytical, as well as automated systems in order to create new schemes and algorithms, their adaptation to non-stereotypical conditions of use, including for the implementation of artificial intelligence methods in the conditions of autonomous work, limitation of computing resources, remote control, etc.

KEYWORDS

Intelligent systems, decision support systems, artificial intelligence, mathematical support.

CIRCLE OF READERS AND SCOPE OF APPLICATION

The monograph will be useful for researchers involved in solving problems of mathematical support for decision-making, using the theory of artificial intelligence, developing new (improving existing) approaches to solving complex technical problems in various fields of human activity.

The monograph is also useful for practitioners – designers, developers implementing modern solutions in the field of information technology, engaged in the development of information, information and analytical, as well as automated control systems in order to create new schemes and algorithms, their adaptation to non-stereotypical conditions of use, including for the implementation of artificial intelligence methods in the conditions of autonomous work, limitation of computing resources, remote control, etc.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

The high dynamism of the development of social processes and phenomena determines the formation of a new system of the worldview of mankind, the modification (change) of the hierarchy of needs and values, and challenges to the pace and quality of development.

Solving complex problems associated with meeting the requirements of our time requires the use of innovative scientific approaches. Today, the use of modern intellectual technologies, such as neural networks, deep learning, and artificial intelligence, is a prerequisite for the proactive development of all spheres of human activity: medicine, technology, business, environmental protection, education, transport and communication, etc. Thus, the intellectualization of technical and managerial systems can be considered one of the key foundations of the new paradigm of science and technology. The phrase "artificial intelligence systems" today is understandable to everyone. The context of this term is associated with such concepts as robotics, forecasting, processing of large information flows, expert systems, diagnostics, smart home or smart tools projects, cyberphysical space and cyberphysical systems, computer translation, etc.

There is a positive dynamics in the development and implementation of artificial intelligence elements in most types of software: mobile applications, information systems, electronic devices, etc.

This process of "intellectualization" allows us to talk about a gradual increase in the intelligence of modern computer systems capable of performing functions that are traditionally considered intellectual: understanding language, logical inference, using the accumulated knowledge, learning, pattern recognition, as well as learning and explaining their decisions.

In the first section of the monograph, a method of self-organization of information networks in conditions of destabilizing influences was proposed. The formalization of the task of managing the resources of the information network to the maximum bandwidth is carried out. In the context of this study, a comprehensive model of the functioning of information networks is proposed, as well as an algorithm for implementing the method of self-organization of information networks in conditions of destabilizing influences.

In the second section, the development of control methods based on bioinspired algorithms was carried out. The proposed methods are based on canonical bioinspired algorithms such as the flying squirrel algorithm, the goose flock algorithm, and the snake flock algorithm. The effectiveness of the proposed bioinspired algorithms was simulated, as well as their comparison with known ones.

The third section is the development of methods for assessing the state of complex technical systems using the theory of artificial intelligence. In this study, the following swarm algorithms were used as the basic ones: a flock of bats algorithm, an invasive weed algorithm, a flock of fish algorithm, as well as a flock of frogs algorithm. The advantages and disadvantages of each approach are determined, and the indicators for assessing the effectiveness of these algorithms with the appropriate justification are determined.

In the fourth section, the development of methods for training artificial neural networks of intelligent decision support systems was investigated. A method for training artificial neural networks, with evolving architecture, a method for training artificial neural networks Kohonen, with evolving architecture, as well as a method for training cascading artificial neural networks, with evolving architecture are proposed. The effectiveness of the proposed methods was evaluated, and their advantages and disadvantages were determined.

The fifth section of the monograph presents a scientific and methodological apparatus for increasing the efficiency of information processing using artificial intelligence. A mathematical formulation of the research problem was carried out with the help of a pack of wolves. A mathematical model of parametric optimization was developed based on an improved algorithm of fireflies in special-purpose information systems. The following scientific results were developed: a method of parametric optimization based on an improved algorithm of a pack of wolves; a method of parametric evaluation of the control object based on an improved firefly algorithm; methods of finding solutions using an improved locust flock algorithm; method of finding solutions using an improved algorithm for emperor penguins. The effectiveness of the proposed scientific results was evaluated, and the advantages and disadvantages of the proposed scientific results were determined.

DOI: 10.15587/978-617-8360-13-9.CH1

Svitlana Kashkevich, Iraida Stanovska, Oleh Shknai,
Oleksandr Lytvynenko, Yevhen Artamonov, Andrii Veretnov

CHAPTER 1

SCIENTIFIC METHOD APPARATUS FOR INTELLECTUAL ASSESSMENT OF THE STATE OF COMPLEX SYSTEMS

ABSTRACT

In this section of the research, a scientific and method apparatus for intelligent assessment of the state of complex systems is proposed. The basis of this research is the theory of artificial intelligence, namely evolving artificial neural networks, basic genetic algorithm procedures, neuro-fuzzy expert systems and bio-inspired algorithms. In the course of the research, the authors proposed:

– the method approach to assessing the state of hierarchical systems using a metaheuristic algorithm;

– the method of analysis and forecasting of the state of multidimensional objects using a metaheuristic algorithm;

– the method of increasing the reliability of the assessment of the object state.

The use of the proposed scientific and method apparatus will allow:

– to reduce the probability of premature convergence of the metaheuristic algorithm;

– to maintain a balance between the speed of convergence of the metaheuristic algorithm and diversification;

– to take into account the type of uncertainty and noisy data of the metaheuristic algorithm;

– to take into account the available computing resources of the state analysis system of the analysis object;

– to take into account the priority of search by swarm agents of the metaheuristic algorithm;

– to carry out the initial display of flock individuals taking into account the uncertainty type;

– to conduct accurate training of individuals of metaheuristic algorithms;

– to conduct a local and global search taking into account the degree of noise of the data on the state of the analysis object;

– to apply as a universal tool for solving the task of analyzing the state of analysis objects due to the hierarchical description of analysis objects;

– to check the reliability of the obtained results;

– to increase the reliability of the assessment of the objects state of analysis due to the construction of object and relational models of their state with different degrees of hierarchy;

– to avoid the local extremum problem.

KEYWORDS

Bio-inspired algorithms, multi-agent systems, combined systems, reliability and efficiency.

1.1 A METHOD APPROACH TO ASSESSING THE STATE OF HIERARCHICAL SYSTEMS USING A METAHEURISTIC ALGORITHM

The process of assessing complex and hierarchical systems is a complex process of determining a set of possible states for a wide range of tasks, including for making management decisions [1–10].

State assessments of complex and hierarchical systems are discontinuous, undifferentiated and multimodal. Considering the above, it is impractical to use classic gradient deterministic algorithms [11–22] to solve this type of problem.

The most common approaches to assessing the state of hierarchical systems are swarm intelligence algorithms (swarm algorithms). The most famous swarm algorithms are particle swarm optimization algorithm, artificial bee colony algorithm, firefly swarm algorithm, ant colony optimization algorithm, wolf swarm optimization algorithm and sparrow swarm algorithm [23–40].

However, most of the basic bio-inspired algorithms mentioned above are unable to maintain a balance between research and use, resulting in unsatisfactory performance for real-world complex optimization tasks.

This encourages the implementation of various strategies to improve the convergence speed and accuracy of the underlying bio-inspired algorithms. Therefore, research devoted to the development of new approaches to assessing the state of complex hierarchical systems is relevant.

An analysis of works [41–71] showed that the common shortcomings of the above-mentioned researches are:

– no possibility of hierarchical processing of various data types;

– the lack of possibility of additional involvement of necessary computing resources of the system;

– a failure to take into account the type of uncertainty and noisy data about the information circulating in the system;

– the lack of deep learning mechanisms of knowledge bases;

– the lack of search priority in a certain direction.

The aim of the research is the development of method approach to assessing the state of hierarchical systems using a metaheuristic algorithm. This will allow to increase the efficiency of assessment of the state of hierarchical systems with a given reliability and the development of subsequent management decisions. This will make it possible to develop software for intelligent decision-making support systems.

To achieve the aim, the following tasks were set:

– to determine the procedures for implementing a method approach to assessing the state of hierarchical systems;

– to lead an example of assessing hierarchical systems while analyzing the operational situation of a group of troops (forces) using the proposed method approach.

In this research, an optimizer based on simulating the behavior of antlions is proposed – a population-based stochastic algorithm that uses antlion agents (ALA) as search agents. The antlion algorithm is based on the imitation of the way antlions dig anthills to hunt ants in the natural environment.

The method approach to assessing the state of hierarchical systems using the metaheuristic algorithm consists of the following sequence of actions:

Step 1. Input of initial data. At this stage, the main parameters of the algorithm are determined, such as:

– the type of task being solved;

– the number of agents in the population;

– the number of variables characterizing the task being solved;

– available computing resources of the system;

– the complexity of the hierarchical system to be assessed;

– the parameters of the improved genetic algorithm (selection parameters, mutations), the number of individuals;

– the type of uncertainty about the hierarchical system (complete uncertainty, partial uncertainty, complete awareness);

– volume and type of research sample;

– volume and type of test sample;

– artificial neural network architecture, etc.

Step 2. Creation of ALA flock. Initialization of the ALA population X_i ($i=1, 2, ..., n$) takes place. The set of ALA form a population, which is described by the matrix X. The initial population of ALA in this algorithm is generated taking into account the uncertainty about the state of the hierarchical system based on the constraints of the problem under consideration. The members of the ALA population are search agents in the solution space, providing candidate values for the problem variables based on their positions in the search space. Mathematically, each member of the general set is a vector, the number of elements of which is equal to the number of task variables.

ALA is issued taking into account the uncertainty about the state of a complex hierarchical system based on basic system models and circulating data models [2, 19, 21] (1.1):

$$X=\begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1}\times\iota_{1,1} & \cdots & x_{1,d}\times\iota_{1,d} & \cdots & x_{1,m}\times\iota_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1}\times\iota_{i,1} & \cdots & x_{i,d}\times\iota_{i,d} & \cdots & x_{i,m}\times\iota_{i,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1}\times\iota_{N,1} & \cdots & x_{N,d}\times\iota_{N,d} & \cdots & x_{N,m}\times\iota_{N,m} \end{bmatrix}_{N\times m}. \quad (1.1)$$

where X is the ALA population matrix; X_i is the i-th member of the ALA swarm (solution candidate); $x_{i,d}$ is the d-th dimension in the search space (decision variable); N is the number of ALA; m is the number of decision variables describing the state complex hierarchical system.

Step 3. Numbering of ALA in the flock, i, $i \in [0, S]$. At this stage, each ALA is assigned a serial number. This makes it possible to determine the parameters of finding a solution for each individual in the flock.

Step 4. Determination of the initial ALA speed.

Initial speed v_0 of each ALA is defined by the following expression:

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

The process of updating the ALA population is based on the simulation of two strategies of the exploration phase and the exploitation phase.

Step 5. Validation of each ALA.

The relevance of each search ALA is determined in each iteration using the improved genetic algorithm proposed in work [26] and comparison of the obtained values with standardized functions. The fitness value of each ALA in the search swarm (each row in the X matrix) is measured and compared with the fitness of the remaining ALA (the other rows of the X matrix).

Step 6. Preliminary assessment of the ALA search area. In this procedure, the natural language search area is determined precisely by the halo of the existence of the ALA, where the ants live.

Step 7. Classification of ant nests.

The location of the best anthill (thus, the smallest anthill with the least number of ants) is considered to be (FS_{ht}), which is nearby and requires the least amount of energy to find and retrieve it. The largest anthill, with the largest number of ants, will be denoted as FS_{at}.

Other single ants will be denoted as FS_{nt}:

$$FS_{ht} = FS(\text{sorte\_index}(0,8)), \tag{1.3}$$

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

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

Step 8. Determining the number of available computing resources of the system.

At this stage, the amount of computing resources available for calculations is determined. In accordance with the provisions outlined in Step 4, the concept of updating the provisions of the ALA is chosen.

Step 9. Reconnaissance (surrounding the prey).

The position of ALA is directly dependent on the position of their prey (in our case, these are ants). The position of each ant in each dimension is updated using a random walk. This random walk is described by the following mathematical expression:

$$x(t) = \left[0, cumsum\left(2t(t_1) - 1\right), cumsum\left(2t(t_2) - 1\right), \ldots, cumsum\left(2t(t_T) - 1\right)\right], \tag{1.6}$$

where T is the maximum number of iterations; t_i is the t-th iteration; *cumsum* is the cumulative summation; $r(t)$ is a random function calculated as follows:

$$r(t) = \begin{cases} 1, & rand \geq 0.5; \\ 0, & rand < 0.5, \end{cases} \tag{1.7}$$

where t is the iteration index; *rand* is a randomly generated number in [0, 1].

The total population of ants on the search plane, on which ALA hunting takes place, is described by the matrix:

$$M_{ant} = \begin{bmatrix} \overrightarrow{Ant_1} \\ \overrightarrow{Ant_2} \\ . \\ . \\ \overrightarrow{Ant_n} \end{bmatrix}, \tag{1.8}$$

where n is the number of ants in the population.

The value of anthills in this research is identified as the value of the decision made in relation to the optimization task, stored in the following vector:

$$M_{oa} = \begin{bmatrix} f\left(\overrightarrow{Ant_1}\right) \\ f\left(\overrightarrow{Ant_2}\right) \\ . \\ . \\ . \\ f\left(\overrightarrow{Ant_n}\right) \end{bmatrix}, \tag{1.9}$$

Each antlion is represented by a pose vector and a target vector as follows:

$$\overrightarrow{Antlion_i} = \left[A_{i,1}, A_{i,1}, \ldots, A_{i,d}\right], \tag{1.10}$$

where $Antlion_i$ is the i-th ant lion; $A_{i,d}$ is the position of the i-th ant in the d-th dimension.

$$M_{Antlion} = \begin{bmatrix} \overrightarrow{Antlion_1} \\ \overrightarrow{Antlion_2} \\ . \\ . \\ \overrightarrow{Antlion_n} \end{bmatrix}, \tag{1.11}$$

$$M_{oal} = \begin{bmatrix} f\left(\overrightarrow{Antlion_1}\right) \\ f\left(\overrightarrow{Antlion_2}\right) \\ . \\ . \\ . \\ f\left(\overrightarrow{Antlion_n}\right) \end{bmatrix}, \tag{1.12}$$

where n is the number of ants in the population.

Step 10. Verification of hitting the global optimum. At this stage, the condition for the algorithm to reach the global optimum is checked according to the specified criterion for assessing the state of complex hierarchical systems.

Step 11. Global restart procedure.

The restart procedure can effectively improve the ability of the algorithm to go beyond the current optimum and improve the exploratory ability of the algorithm. If the optimal population of the algorithm remains unchanged after T iterations, the population is likely to fall into a local optimum. Thus, the candidate solution will be initialized randomly to accelerate the departure from the global optimum.

Step 12. Hunting phase (exploitation).

To determine the priority of the anthill, the anthill with the highest value of ant pheromone (with more ants) is chosen for the ALA attack:

$$P_{ij}^{k} = \begin{cases} \dfrac{\left(\tau_{ij}\right)^{\alpha}\left(\eta_{ij}\right)^{\beta}}{\sum\limits_{h \notin tabu_k}\left(\tau_{ih}\right)^{\alpha}\left(\eta_{ih}\right)^{\beta}}, & j \notin tabu_k; \\ 0, & \text{otherwise}, \end{cases} \tag{1.13}$$

where τ_{ij} and η_{ij} is the intensity of pheromones and the cost of the route between anthills i and j, respectively. Relative value τ_{ij} and η_{ij} is determined by the parameters α and β, respectively. $tabu_k$ is a list of unavailable routes (visited nodes) for ALA k.

Step 13. Checking the stop criterion. The algorithm terminates when the maximum number of iterations is completed. Otherwise, the behavior of generating new places and checking conditions is repeated.

Step 14. Learning ALA knowledge bases.

In this research, the learning method based on evolving artificial neural networks developed in the research [2] is used to train the knowledge bases of each ALA. The method is used to change the nature of movement of each ALA, for more accurate analysis results in the future.

The end of algorithm.

A method approach to assessing the state of hierarchical systems using a metaheuristic algorithm is proposed. To determine the effectiveness of the proposed methodical approach, modeling of its work was carried out to solve the task of determining the composition of the operational grouping of troops (forces) and the elements of its operational construction in order to determine the expediency of regrouping troops (forces).

The effectiveness of the method approach is compared with swarm optimization algorithms, using a set of CEC2019 test functions listed in the **Table 1.1**. The efficiency assessment criterion is the speed of decision making (msec) with a given assessment reliability (0.9).

As it can be seen from the **Table 1.1**, increasing the efficiency of assessing the state of hierarchical systems is achieved at the level of 22–25 % due to the use of additional procedures.

It can be seen that the method approach is able to converge to the true value for most unimodal functions with the fastest convergence speed and the highest accuracy, while the convergence results of the ant swarm algorithm are far from satisfactory.

The advantages of the proposed method approach are due to the following:

– the initial position of the ALA is carried out taking into account the type of uncertainty (Step 2) due to the use of appropriate correction coefficients for the degree of awareness of the placement of anthills (in our case, priority search directions), in comparison with works [9, 14, 21];

– the initial speed of each ALA is taken into account (Step 4), which allows to determine the search priority of each ALA in the specified search direction, in comparison with works [9–15];

– the suitability of ALA hunting sites is determined, which reduces the time for assessing the state of the hierarchical system (Step 6), in comparison with works [14, 16, 17];

– the degree of data noise is taken into account in the process of updating the ALA position (Steps 9–12), thereby reducing the time for assessing the state of hierarchical systems, compared to works [9–15];

– the use of the procedure of global restart of the algorithm, which achieves the ability of the algorithm to go beyond the current optimum and improve the research ability of the algorithm (Step 11), which reduces the time for assessing the state of hierarchical systems, compared to works [9–15];

– the universality of solving the task of assessing the state of hierarchical ALA systems due to the hierarchical nature of their description (Steps 1–14, **Table 1.1**), in comparison with works [9, 12–18];

– the possibility of simultaneously searching for a solution in different directions (Steps 1–14, **Table 1.1**);

– the adequacy of the obtained results (Steps 1–14), in comparison with works [9–23];

– the possibility of clarifying the selection of an anthill at the hunting stage (Step 12) due to the ranking of anthills by the level of ant pheromone, in comparison with works [9, 12–18];

– an improved possibility of selecting the best ALA in comparison with random selection due to the use of an improved genetic algorithm (Step 5), in comparison with works [9–15]. This allows to improve the reliability of assessment of the state of hierarchical systems;

– the ability to avoid the local extremum problem (Steps 1–14);

– the possibility of in-depth learning of ALA knowledge bases (Step 14), in comparison with works [9–23].

CHAPTER 1

Table 1.1 Comparison of the proposed method approach with other swarm algorithms for a defined set of test functions

Type of test functions CEC2019	Value	Particle swarm algorithm	Ant colony algorithm	Black widow algorithm	Algorithm of a flock of gray wolves	Bee swarm algorithm	Canonical algorithm of ant lions	The proposed method approach
F1	Better	6.2501	4.4884	4.103	4.4136	6.2606	5.0994	**2.7698**
	Average	8.1507	6.2966	5.309	5.1521	6.2606	8.1594	**2.7698**
	Standard	7.33	6.01	6.2	6.4	7.838	7.7192	**3.3712**
F2	Better	545.9192	5.6911	4.8556	4.2197	4.0557	4.2739	**3.2141**
	Average	2689.105	6.91	4.9935	55.3157	4.8087	4.9449	**4.6568**
	Standard	1741.300	0.94	0.02708	106.434	0.33374	0.17328	**0.64381**
F3	Better	2.3979	2.4361	1.9805	1.1634	1.4104	2.9411	**1.4173**
	Average	7.2501	4.4884	4.103	4.4136	6.2606	5.0994	**2.7698**
	Standard	1.9616	1.2868	0.766	2.2871	2.6603	1.0752	**0.62451**
F4	Better	9.2209	10.0576	38.8009	12.248	4.1414	29.1901	**3.9849**
	Average	27.9446	25.5342	57.3153	24,843	28.4397	44.8217	**16.6629**
	Standard	11.0734	8.7901	6.9365	10.7428	15.7463	8.8591	**11.1345**
F5	Better	1.5511	1.4833	29.405	1.1339	1.0497	2.1918	**1.0074**
	Average	11.428	1.7597	72.0211	9.5542	1.115	3.7237	**1.0641**
	Standard	13.5064	0.18663	19.4782	9.0179	0.061305	1.1233	**0.050987**
F6	Better	1.9552	3.1214	8.5015	1.7038	1.3443	6.2846	**1.0041**
	Average	6.7367	5.9982	10.5902	5.2258	3.9784	9.1707	**3.0444**
	Standard	2.4593	1.2484	0.77804	1.7237	1.5453	1.3384	**1.268**
F7	Better	308.3668	249.343	1208.30	309.912	83.4932	399.241	**126.6386**
	Average	1102.474	832.301	1623.67	810.994	894.644	1206.1076	**508.5085**
	Standard	355.712	265.703	130.086	376.011	339.8851	290.3894	**230.6677**
F8	Better	805.925	811.990	837.627	809.783	804.9748	816.6299	**802.0457**
	Average	825.7583	824.501	848.282	824.769	825.9996	832.6092	**815.3867**
	Standard	10.1005	7.4753	6.0262	9.2526	8.9454	8.4370	**9.9757**
F9	Better	1.1616	1.1532	1.5475	1.2025	1.1683	1.3514	**1.0359**
	Average	1.3579	1.4924	1.9341	1.3228	1.3786	1.6591	**1.1378**
	Standard	0.0999	0.1697	0.1336	0.0928	0.1442	0.1545	**0.0522**
F10	Better	1436.99	1274.08	1641.75	1325.81	1204.19	1818.69	**1139.99**
	Average	1937.94	1861.89	2362.49	1828.64	1819.19	2217.23	**1505.39**
	Standard	349.35	275.94	194.21	344.89	267.68	216.92	**229.69**

The disadvantages of the proposed method approach should include:

– the loss of informativeness while processing various types of data due to the construction of the membership function;

– lower accuracy of processing one type of data due to gradient search;

– the loss of credibility of the obtained solutions while searching for a solution in several directions at the same time;

– lower accuracy of assessment compared to other assessment approaches.

The specified method approach will allow:

– to assess the state of complex hierarchical systems;

– to determine effective measures to improve the efficiency of assessing the state of complex hierarchical systems while maintaining the given reliability;

– to reduce the use of computing resources of decision-making support systems.

The limitations of the research are the need to have an initial database on the state of hierarchical systems, the need to take into account the time delay for collection and proving information from intelligence sources.

It is advisable to use the proposed method approach to solve the problems of assessing the state of complex hierarchical systems in conditions of uncertainty and risks characterized by a high degree of complexity.

This research is a further development of researches aimed at developing method principles for increasing the efficiency of processing various types of data, which were published earlier [2, 4–6, 21–23].

The directions of further research should be aimed at reducing computing costs when processing various types of data in special purpose systems.

1.2 THE DEVELOPMENT OF A METHOD FOR ANALYZING AND FORECASTING THE STATE OF MULTIDIMENSIONAL OBJECTS USING A METAHEURISTIC ALGORITHM

The Butterfly Swarm Algorithm (BSA) is based on the behavior of a swarm of butterflies in search of food. As a rule, butterflies can determine the source of the aroma accurately and distinguish between different aromas. Butterflies move from their place to other places with more nectar. Butterflies produce scent as they move to share their current location and personal information with other butterflies.

The inspiration and behavior of butterflies can in this research be formulated as an optimization method, where butterflies represent search agents and produced aromas characterize the value of optimization.

In BSA, butterfly agents (BA) can generate a scent/fitness value with some strength to distinguish it from other scents. This behavior can assist other BA in updating their position in the search space. Once BA that finds the best nectar in the search space produces a scent, all neighboring BA move to the best location for the BA. This kind of mechanism update is called a global search in BSA.

On the other hand, BA will move randomly in the search space if the scents of other BA are detected, known as local search.

The method of analyzing and forecasting the state of multidimensional objects using the metaheuristic algorithm consists of the following sequence of actions:

Step 1. Input of initial data. At this stage, the main parameters of the algorithm are determined, such as:

– the type of task being solved;

– the number of BA in the population;

– the number of variables characterizing the task of analysis and forecasting of multidimensional objects to be solved;

– available computing resources of the system of analysis and forecasting of multidimensional objects;

– the complexity of multidimensional objects to be assessed;

– the parameters of the improved genetic algorithm (selection parameters, mutations), the number of individuals to be selected;

– the type of uncertainty about the state of multidimensional objects (complete uncertainty, partial uncertainty and complete awareness);

– the volume and type of training sample for artificial neural networks;

– the volume and type of test sample for artificial neural networks;

– artificial neural network architecture, etc.

Step 2. Creation of a BA flock. Initialization of the primary (initial) BA population X_i ($i=1, 2, ..., n$) takes place. All BAs form a population (flock), which is determined by the matrix X. The initial population (flock) of BA in this research is generated taking into account the uncertainty about the state of multidimensional objects based on the constraints of the problem under consideration (analysis and/or prediction):

$$X=\begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m}=\begin{bmatrix} x_{1,1}\times\iota_{1,1} & \cdots & x_{1,d}\times\iota_{1,d} & \cdots & x_{1,m}\times\iota_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1}\times\iota_{i,1} & \cdots & x_{i,d}\times\iota_{i,d} & \cdots & x_{i,m}\times\iota_{i,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1}\times\iota_{N,1} & \cdots & x_{N,d}\times\iota_{N,d} & \cdots & x_{N,m}\times\iota_{N,m} \end{bmatrix}_{N\times m}. \quad (1.14)$$

where X is the matrix describing the BA population on the problem solving plane; X_i is the i-th BA that is a solution candidate; $x_{i,d}$ is the d-th dimension of the multidimensional object in the solution search space; N is the number of BA in the population (flocks); m is the number of decision variables describing the state of a multidimensional object.

Step 3. Assigning a serial number to each BA in the flock, i, $i \in [0, S]$. This step allows to determine the parameters of finding a solution for each BA in the population.

Step 4. Setting the initial speed of each BA.

Initial speed for an individual BA v_0 is described by the following mathematical expression:

$$v_i = \left(v_1, v_2, \ldots, v_S\right), v_i = v_0. \quad (1.15)$$

The population update of each BA in this research is determined in order to determine the balance between the two major procedures of exploration and exploitation. These procedures define the global and local search process. The process of updating the BA population is based on the simulation of two strategies of the exploration phase and the exploitation phase.

Step 5. Selection of BA for solving tasks.

The suitability for solving tasks of each BA is determined in each iteration using the improved genetic algorithm proposed in work [26] and comparing the obtained values with standardized functions. The fitness of the BA in the search flock (row in the X matrix) is measured and compared with the fitness of the rest of the BA (the other rows of the X matrix).

Step 6. Preliminary assessment of the BA search area. In this procedure, the search area in natural language is determined precisely by the halo of BA existence, where butterflies live.

Step 7. Classification of nectar sources for BA.

The location of the best cluster of nectar sources for BA is considered to be (FS_{ht}), which is nearby and requires the least amount of energy to find and collect it. Let's denote the most distant clusters of nectar sources as FS_{at}.

Other single sources of nectar will be denoted as FS_{nt}:

$$FS_{ht} = FS(\text{sorte\_index}(0,7)), \quad (1.16)$$

$$FS_{at}(1:3) = FS(\text{sorte\_index}(1:4)), \quad (1.17)$$

$$FS_{nt}(1:NP\text{-}4) = FS(\text{sorte\_index}(2:NP)). \quad (1.18)$$

Step 8. Determining the intensity of the aroma of nectar sources.

The aroma intensity for BA is mathematically modeled as follows:

$$pf_i = cI^a, \quad (1.19)$$

where pf_i is the aroma strength from the i-th BA; I is the stimulus intensity; c is the sensory modality; a is an indicator of the degree of dependence on the modality.

Step 9. Determination of the number of available computing resources of the system.

At this stage, the amount of computing resources available for calculations is determined. In accordance with the provisions outlined in Step 4, the concept of updating the BA provision is chosen.

Step 10. Exploration of nectar sources (global search).

The location of each BA is represented as a vector of certain problem values. This BA location can be updated by trying to find a better location using the following formula:

$$x_i^{t+1} = x_i^t + F_i^{t+1}, \tag{1.20}$$

where x_i^t is the current position of BA i in iteration t; x_i^{t+1} is the following position of the i-th BA; F_i^{t+1} is the fragrance x_i used to update its position during iterations.

In the global search, the i-th BA moves to the strongest BA g^*, which can be represented as:

$$F_i^{t+1} = \left(r^2 \times g^* - x_i^t\right) \times pf_i, \tag{1.21}$$

where r is a random number in the range [0, 1].

Step 11. Verification of hitting the global optimum. At this stage, the condition of the algorithm hitting the global optimum is checked according to the defined criterion for assessing the state of the multidimensional object.

Step 12. Global restart procedure.

The restart procedure can effectively improve the ability of the algorithm to go beyond the current optimum and improve the exploratory ability of the algorithm. If the optimal population of the algorithm remains unchanged after T iterations, the population is likely to fall into a local optimum. Thus, the candidate solution will be initialized randomly to accelerate the departure from the global optimum.

Step 13. Nectar extraction phase (exploitation).

In the local search, the movement of BA updates can be formulated as follows:

$$F_i^{t+1} = \left(r^2 \times x_j^t - x_k^t\right) \times pf_i, \tag{1.23}$$

where x_j^t and x_k^t is the position of the j-th and k-th BA in the search area. A new parameter, called the switching probability p, is used in BSA to switch the behavior of the algorithm between local and global search to obtain the best balance between exploration and exploitation.

Step 14. Checking the stop criterion. The algorithm terminates if the maximum number of iterations is completed. Otherwise, the behavior of generating new places and checking conditions is repeated.

Step 15. Training of BA knowledge bases.

In this research, the learning method based on evolving artificial neural networks developed in the research [2] is used to learn the knowledge bases of each BA. The method is used to change the nature of movement of each BA, for more accurate analysis results in the future.

The end of algorithm.

This section analyzes the BA behavior in exploration and exploitation due to its significant impact on the solution of analysis and forecasting tasks. The success of metaheuristics lies in their ability to achieve the best balance between exploration and exploitation. These two terms conflict in their search behavior. At the research stage, the algorithm has a high ability to explore and move through unexplored areas of the search space, while at the exploitation stage, the algorithm processes focus on deep search in known areas of the search space.

To determine the effectiveness of the proposed method for analyzing and forecasting the state of multidimensional objects using a metaheuristic algorithm, a simulation of its work was carried out to solve the task of determining the composition of an operational grouping of troops (forces) and elements of its operational construction in order to determine the expediency of regrouping troops (forces).

The effectiveness of the method is compared with metaheuristic algorithms, using a set of CEC2017 test functions listed in the **Table 1.2**. The criterion for assessing efficiency is the speed of decision making (msec) with the given reliability of the evaluation (0.9).

Table 1.2 Comparison of the proposed method with other metaheuristic algorithms for a defined set of test functions

The type of test functions	Metrics	Particle swarm algorithm	Ant colony algorithm	Black widow algorithm	Algorithm of a flock of gray wolves	Bee swarm algorithm	Canonical butterfly swarm algorithm	The proposed method
1	2	3	4	5	6	7	8	9
CEC 2017-F1	Average	3.61E+09	7.21 E+07	3.54E+09	3.88E+09	7.98E+09	5.38E+09	**2.36E+05**
	Standard	3.15E+09	1.21E+08	1.70E+09	2.37E+09	5.03E+09	3.18E+09	**8.39E+04**
CEC 2017-F2	Average	8.46E+31	4.75E+29	3.91E+31	1.44E+35	3.21E+34	7.36E+34	**1.97E+33**
	Standard	4.49E+32	2.54E+30	1.53E+32	5.49E+35	1.75E+35	3.95E+35	**1.08E+34**
CEC 20l7-F3	Average	1.57E+05	1.11E+05	5.58E+04	7.19E+04	6.82E+04	6.87E+04	**2.12E+04**
	Standard	5.23E+04	3.72E+04	1.02E+04	1.46E+04	2.04E+04	1.46E+04	**1.05E+04**
CEC 2017-F4	Average	8.33E+02	6.40E+02	1.02E+03	7.45E+02	1.22E+03	9.81E+02	**6.99E+02**
	Standard	1.86E+02	5.06E+01	2.56E+02	1.57E+02	7.74E+02	4.35E+02	**2.34E+02**
CEC 2017-F5	Average	7.35E+02	7.07E+02	7.19E+02	6.50E+02	6.72E+02	6.36E+02	**6.37E+02**
	Standard	2.49E+01	3.09E+01	3.47E+01	4.17E+01	3.66E+01	4.20E+01	**2.09E+01**
CEC 2017-F6	Average	6.57E+02	6.56E+02	6.38E+02	6.16E+02	6.32E+02	6.23E+02	**6.08E+02**
	Standard	1.1 lE+01	8.62E+00	1.03E+01	5.22E+(X)	1.01E+01	8.22E+00	**6.15E+(X)**
CEC 2017-F7	Average	1.26E+03	1.14E+03	1.31E+03	9.24E+02	9.83E+02	9.26E+02	**8.78E+02**
	Standard	7.14E+01	7.87E+01	1.30E+02	8.40E+01	5.10E+01	5.80E+01	**3.01E+01**
CEC 2017-F8	Average	9.65E+02	9.63E+02	1.00E+03	9.30E+02	9.48E+02	9.18E+02	**9.00E+02**
	Standard	1.82E+01	2.78E+01	3.37E+01	4.64E+01	2.80E+01	2.67E+01	**1.97E+01**

Continuation of Table 1.2

1	2	3	4	5	6	7	8	9
CEC 2017-F9	Average	8.82E+03	6.16E+03	6.71E+03	3.55E+03	5.40E+03	4.06E+03	**3.18E+03**
	Standard	1.80E+03	9.63E+02	1.33E+03	1.36E+03	1.53E+03	1.54E+03	**7.41E+02**
CEC 2017-F10	Average	7.54E+03	5.19E+03	8.27E+03	4.90E+03	4.64E+03	4.74E+03	**3.52E+03**
	Standard	4.70E+02	7.04E+02	3.05E+02	1.46E+03	1.15E+03	1.59E+03	**6.20E+02**
CEC 2017-F11	Average	3.48E+03	2.53E+03	2.20E+03	6.68E+03	8.37E+03	1.19E+04	**131E+03**
	Standard	1.35E+03	1.10E+03	9.02E+02	4.18E+03	4.34E+03	6.41E+03	**1.88E+02**
CEC 2017-F12	Average	1.80E+09	7.55E+07	1.62E+09	1.16E+09	5.92E+08	5.87E+07	**6.77E+07**
	Standard	9.48E+08	3.90E+07	9.40E+08	6.26E+08	1.96E+09	8.65E+07	**5.88E+07**
CEC 2017-F13	Average	7.05E+07	2.18E+04	1.15E+08	8.35E+07	4.58E+04	8.45E+04	**7.63E+04**
	Standard	1.39E+08	2.43E+04	1.36E+08	2.11E+08	4.80E+04	6.59E+04	**7.05E+04**
CEC 2017-F14	Average	3.06E+06	3.07E+05	1.10E+06	1.04E+06	4.25E+05	5.37E+05	**3.55E+05**
	Standard	4.63E+06	3.06E+05	1.07E+06	1.07E+06	5.26E+05	5.94E+05	**3.42E+05**
CEC 2017-F15	Average	8.43E+03	2.81E+05	1.05E+06	2.28E+05	2.96E+04	4.87E+04	**5.04E+04**
	Standard	3.08E+03	1.57E+05	1.10E+06	1.29E+05	2.27E+04	4.26E+04	**5.56E+04**
CEC 2017-F16	Average	3.60E+03	3.56E+03	4.12E+03	3.86E+03	3.02E+03	3.05E+03	**3.19E+03**
	Standard	4.40E+02	1.90E+02	8.26E+02	5.61E+02	4.96E+02	4.26E+02	**4.50E+02**
CEC 2017-F17	Average	2.82E+03	2.58E+03	2.90E+03	3.04E+03	2.79E+03	2.34E+03	**2.67E+03**
	Standard	2.71E+02	1.65E+02	3.62E+02	3.13E+02	3.04E+02	1.97E+02	**3.08E+02**
CEC 2017-F18	Average	4.28E+06	4.65E+06	3.70E+07	1.15E+07	1.61E+06	1.84E+06	**1.83E+06**
	Standard	3.92E+06	2.52E+06	4.64E+07	8.25E+06	1.44E+06	1.40E+06	**1.56E+06**
CEC 2017-F19	Average	9.03E+05	3.77E+03	2.12E+07	7.52E+06	3.31E+05	1.30E+05	**4.71E+05**
	Standard	1.08E+06	2.05E+03	1.84E+07	5.42E+06	9.21E+05	2.12E+05	**5.42E+05**
CEC 2017-F20	Average	3.26E+03	2.99E+03	2.94E+03	2.96E+03	3.00E+03	2.85E+03	**2.79E+03**
	Standard	2.52E+02	1.10E+02	2.03E+02	1.99E+02	2.47E+02	2.11E+02	**1.86E+02**
CEC 2017-F21	Average	2.20E+03	2.20E+03	2.20E+03	2.20E+03	2.20E+03	2.20E+03	**2.20E+03**
	Standard	1.60E+00	3.25E-06	2.98E-02	1.15E+00	1.49E-01	5.43E-03	**1.26E-03**

As it can be seen from the **Table 1.2**, increasing the efficiency of assessing the state of hierarchical systems is achieved at the level of 14–16 % due to the use of additional procedures.

The method converges to the true value for most unimodal functions with the fastest convergence speed and the highest accuracy, while the convergence results of the black widow algorithm are far from satisfactory.

The advantages of the proposed method are due to the following:

– the initial display of BA on the plane of multidimensional objects is carried out taking into account the type of uncertainty (Step 2) due to the use of appropriate correction coefficients for the degree of awareness of the location of nectar sources (in our case, priority search directions), in comparison with works [9, 14, 21];

– adjusting the initial speed of the BA (Step 4) allows to determine the priority of the search, in comparison with works [9–15];

– the suitability of BA nectar collection sites is determined, which reduces the time for assessing the state of multidimensional objects (Step 6), in comparison with works [14, 16, 17];

– the presence of the possibility of global restart of the algorithm, which enables the algorithm to go beyond the current optimum and improve the algorithm's research ability (Step 11), which reduces the time needed to assess the state of multidimensional objects, compared to works [9–15];

– the universality of solving the task of assessing the state of hierarchical BA systems due to the hierarchical nature of their description (Steps 1–15, **Table 1.2**), in comparison with works [9, 12–18];

– the possibility of simultaneously searching for a solution in different directions (Steps 1–15, **Table 1.2**);

– the adequacy of the obtained results (Steps 1–15), in comparison with works [9–23];

– the possibility of clarification at the stage of collecting nectar clusters (Step 12) due to the ranking of nectar sources by the level of stimulus intensity, in comparison with works [9, 12–18];

– an improved possibility of selecting the best BA in comparison with traditional selection due to the use of an improved genetic algorithm (Step 5), in comparison with works [9–15];

– the ability to avoid the local extremum problem (Steps 1–15);

– the possibility of in-depth learning of BA knowledge bases (Step 15), in comparison with works [9–23].

The disadvantages of the proposed method of approach should include:

– lower accuracy of finding solutions in one direction due to gradient search;

– the loss of credibility of the obtained solutions while searching for a solution in several directions at the same time;

– lower accuracy of assessment compared to other assessment approaches.

This method will allow:

– to assess the condition of multidimensional objects;

– to determine effective measures to improve the efficiency of assessment of the state of multidimensional objects while maintaining the given reliability;

– to reduce the use of computing resources of decision-making support systems.

The limitations of the research are the need to have an initial database on the state of multidimensional objects, the need to take into account the delay time for collection and proving information from intelligence sources.

1.3 DEVELOPMENT OF A METHOD OF INCREASING THE RELIABILITY OF THE ASSESSMENT OF THE OBJECT STATE

The Rete II method [5, 10, 11] was chosen as the basis for the development of the method of increasing the reliability of the assessment of the object state. Despite the advantages of the Rete II method, including such as increased performance of processing various types of data, the presence of a reverse inference algorithm, the main disadvantages of this method are:

– the work only with clear products;

– low speed of setting up databases;

– compared to other methods, the reliability of the assessment is low, which does not allow it to be used in the processing of various types of data that require a high degree of reliability of decision making.

The essence of the proposed method is the following:

– construction of an object model for the assessment of the object state;

– adjusting the neuro-fuzzy knowledge base by several bio-inspired algorithms and combining the results of parallel work of bio-inspired algorithms using the Merge function;

– construction of a relational model of object state assessment.

Due to this set of procedures, an increase in the reliability of the assessment of the object state is achieved.

The algorithm for the implementation of the proposed method of increasing the reliability of the assessment of the object state consists of the following sequence of actions:

Step 1. Input of initial data.

At this stage, the initial data about the object to be assessed is entered. The starting point about the degree of uncertainty about the object state is determined. The initial data for the work of bio-inspired algorithms are determined, the reliability parameters of the research object are introduced.

Step 2. Formation of the object model.

A formal object model using neuro-fuzzy expert systems has the following form:

$$\{P_o\} = \{\mathrm{Rule}_o\}, \tag{1.24}$$

where Rule is a set of rules characterizing the object model of the specified analysis object.

Each Rule is described as follows:

$$\mathrm{Rule}_o = < C \rightarrow S >, \tag{1.25}$$

where C is the condition of each rule of the object model, S is the consequence for each rule of the object model.

For this type of object model, it is necessary to correctly present the grammatical structure of the rules with various types of nested conditions. For this purpose, it is proposed to use a recursive mechanism for describing the nodes and terminal vertices of the rule condition tree. In expression (1.25), C is described as:

$$C = < C_l, R, C_r >, \tag{1.26}$$

where C_l is the left node of the condition of each rule of the object model; R is the relationship between the nodes of each rule of the object model; C_r is the right node of the condition of each rule of the object model. Then, let's consider the given parameters:

$$C_l = FC_l \| \mathrm{Null} \| C, \tag{1.27}$$

$$C_r = FC_r \| \mathrm{Null} \| C, \tag{1.28}$$

where FC_l is the left terminal triple of the condition of each rule of the object model; FC_r is the right final triple of the condition of each rule of the object model. The expressions (1.27) and (1.28) allow to describe the conditions of each rule of the object model with a different degree of hierarchy:

$$FC_l = < L, Z, W >, \tag{1.29}$$

$$FC_r = < L, Z, W >, \tag{1.30}$$

where L is a linguistic variable of the object model; Z is a condition sign $Z = \{<,>,<=,>=,=,!=\}$; W is the condition value of the object model, which is determined as follows (1.31):

$$W = L \| V, \tag{1.31}$$

where L is a linguistic variable of the object model; V is some fixed value (1.32):

$$V = T_i \| \mathrm{const}, \tag{1.32}$$

where T_i is the value of the fuzzy variable from the term-sets of the linguistic variable, const is a constant. The given set of mathematical expressions (1.24)–(1.31) allows the use of not only linguistic variables, but also classical variables.

Similarly to parameter C of the object model, parameter S is defined as a consequence of the rule of the object model:

$$S = < S_l, R, S_r >, \tag{1.33}$$

where S_l is the left node of the consequence of the rule of the object model; R is the relationship between the nodes of the consequence of the rule; S_r is the right node of the consequence of the rule of the object model:

$$S_l = FS_l \|\text{Null}\| S, \tag{1.34}$$

$$S_r = FS_r \|\text{Null}\| S, \tag{1.35}$$

where FS_l is the left terminal triplet of the consequence of the rule of the object model; FS_r is the right final triple of the consequence of the rule of the object model. Formulas (1.34) and (1.35) allow to describe consequences with different degrees of hierarchy:

$$FS_l = < L, \text{Op}, W >, \tag{1.36}$$

$$FS_r = < L, \text{Op}, W >, \tag{1.37}$$

where L is a linguistic variable of the object model; Op is the type of operation, $\text{Op} = \{:=\}$; W is the value of the consequence of the object model.

Step 3. Setting up a neuro-fuzzy knowledge base and combining the results of their work.

At this stage, the results of parallel work of bio-inspired algorithms are combined using the Merge function.

The function element-by-element compares two binary vectors from the outputs of two bio-inspired algorithms is the bat swarm algorithm and the frog swarm algorithm.

Under the condition that the value of the element at the same position coincides, the given value will be written to this position in the resulting vector. Otherwise, a random number is generated from the interval from 0 to 1 [14–20].

If the value is less than or equal to 0.5, then the corresponding position of the new vector is written by the element from the worst vector. Otherwise, an element from a better vector will be displayed at this location.

Thus, the merge function can be given as follows:

$$merge(S_w, S_b) = \begin{cases} s_i^* = s_{wi} = s_{bi}, & \text{if } s_{wi} = s_{bi}; \\ s_i^* = s_{wi}, & \text{if } s_{wi} \neq s_{bi} \text{ and } rand \leq 0.5; \\ s_i^* = s_{bi}, & \text{if } s_{wi} \neq s_{bi} \text{ and } rand > 0.5, \end{cases} \tag{1.38}$$

where $rand$ is a random, uniformly distributed number, $rand \in [0;1]$.

Step 4. Construction of a relational model of object state assessment.

Construction of a relational model of object state assessment in this research is based on the Gray relational analysis (GRA) method. This approach is one of the approaches to multi-criteria analysis, which is used to assess alternatives based on a number of different criteria. This method is used to measure the level of relationship between existing alternatives by calculating the Gray's correlation coefficient (Gray's relational coefficient). The stages of completion by the GRA method are as follows.

Step 4.1. Normalization of relational model data.

Normalization is used to transform data into a single scale that allows better comparison of different variables. The normalization equation in GRA is as follows:

$$X_{norm} = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}}. \tag{1.39}$$

Step 4.2. Formation of the matrix of relational analysis of Gray.

After data normalization, the result of the normalization matrix is the relational analysis of the Gray matrix, namely:

$$G = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix}, \tag{1.40}$$

where G is the result of the data normalization matrix; m is an existing alternative; n is the existing criterion; x_{ij} is a normalization in the measurement of alternatives.

Step 4.3. Multiplying the GRA matrix by weights.

The next step is to determine the relative weight for each variable. This weight reflects the level of importance of each variable in the GRA analysis. In addition, the GRA method is to give each criterion a weighting that relates to the level of importance of the criterion. Below is the formula for calculations:

$$V_{ij} = g_{i,j} w_j. \tag{1.41}$$

Thus, the following results of the weighted normalization matrix can be formed:

$$V = \begin{bmatrix} v_{1,1} & \cdots & v_{1,n} \\ \vdots & \ddots & \vdots \\ v_{m,1} & \cdots & x_{m,n} \end{bmatrix}. \tag{1.42}$$

Step 4.4. Calculating the value of Gray relational analysis.

In this step, the Gray ratio value is calculated for each variable based on the Gray ratio matrix and the relative weights that were determined using the following equation:

$$GRG_i = \frac{1}{n} \sum_{j=1}^{n} V_{ij}, \tag{1.43}$$

where GRG_i is the value of the Gray ratio (GRG) of the i-th variable to the reference variable.

The end of the algorithm.

CHAPTER 1

To determine the effectiveness of the proposed method of increasing the reliability of the assessment of the object state, a simulation of its work was carried out to solve the task of determining the composition of the operational grouping of troops (forces) and the elements of its operational construction in order to determine the feasibility of regrouping troops (forces).

The following linguistic variables were used to solve the problem:

1. Types of radio emitting devices (RED): The range of permissible values: radio communication devices, radio relay devices, satellite communication devices, air monitoring devices (radar detection devices); devices of radio-electronic countermeasures:

RED="Types of radio emitting devices"={"brigade tactical group", "operational grouping of troops (forces)", "strategic grouping of troops (forces)"}.

2. The types of organizational and staff formations: Range of permissible values: 0÷1:

OSF="Types of organizational and staff formations"={"brigade tactical groups", "operational groupings of troops (forces)", "strategic groupings of troops (forces)"}.

3. The types of control points: Range of permissible values: 0÷1:

CP="Types of control points"={"control points of brigade tactical groups", "control points of operational groups of troops (forces)", "control points of strategic groups of troops (forces)"}.

4. The availability of fortifications: The range of permissible values: 0÷1:

F="Availability of fortifications"={"Fortifications of the first tier", "Fortifications of the first and second tiers", "Fortifications of the first to third tiers"}.

5. The availability of reserves: Range of permissible values: 0÷1:

AR="Availability of reserves"={"reserve brigade tactical group", "two reserve brigade tactical groups", "reserve operational group"}.

6. The operation type: The range of permissible values: 0÷1:

TO="Operation type"={" defensive", "offensive", "counter-offensive"}.

7. The activity of actions in the specified direction: The range of permissible values: 0÷1:

AA="Activity of actions in the specified direction"={"low", "medium", "high"}.

8. The uncertainty of operational situation: The range of permissible values: complete uncertainty ÷ complete awareness:

UN="Uncertainty of operational situation"={"Complete uncertainty", "partial uncertainty", "full awareness"}.

To simplify further writing, let's denote the vague variables "zero" is "*Z*", "low" is "*L*", "normal" is "*N*", "high" is "*H*".

The membership functions given in the example for the analysis of the operational situation are presented in the specified form according to the formula:

1) $(CP="H")$ and $(OSF="H")$ and $(UN="H")$ and $(AR="L") \rightarrow (REZ="H")$,

...

81) $(CP="L")$ and $(OSF="L")$ and $(UN="L")$ and $(AR="H") \rightarrow (REZ="L")$,

82) $(F="L")$ and $(AA="L")$ and $(UN="H")$ and $(AR="H") \rightarrow (REZ="N")$,

...

108) $(SC="L")$ and $(OSF="L")$ and $(UN="H")$ and $(CP="L") \rightarrow (REZ = "N")$.

In this example, only a separate part of the rule base of the neuro-fuzzy expert system is given. In the main base of rules there are rules not only with connections of conditions with the help of *T*-norms, but also with the help of *T*-conorms and with negations of conditions.

In the worst case, to find a solution, the system should check all the rules contained in the rule base. Thus, it is necessary to check 617 conditions and calculate 315 *T*-norm operations.

The reliability score for the rule bases (RB*i*) is given in **Table 1.3**.

The classic Rete II, Treat and Leaps method and the proposed method [10–12] were used to compare the reliability of the assessment.

This table clearly shows that the application of the modified Rete II method is justified for rule bases containing a large number of rules and a relatively small number of linguistic variables. In this case, the modified Rete method allows the reliability of information processing to be almost twice that of a fuzzy expert system, and by 20–25 % compared to the classic Rete method.

The research of the developed method showed that the specified method provides an average of 20 % higher reliability of obtaining an estimate (**Table 1.3**).

The advantages of the proposed method are the following:

– the possibility of increasing the reliability of the assessment of the object state due to the parallel use of two bio-inspired algorithms, in comparison with works [9, 12, 19];

– taking into account the degree of awareness of the object state, due to the application of correction coefficients for the degree of awareness, in comparison with works [9–15];

– the construction of both object and relational models, which makes it possible to increase the reliability of the assessment of the objects state, in comparison with works [9–15];

– the possibility of combining the results of the work of bio-inspired algorithms, which makes it possible to mutually check the correctness of the work of each of the algorithms in comparison with works [14, 16, 17];

– universality of solving the task of assessing the condition of objects with different degrees due to the hierarchical nature of their description (Steps 1–4, **Table 1.3**), in comparison with works [9, 12–18];

– the possibility of simultaneously searching for a solution in different directions (Steps 1–4, **Table 1.3**);

– the adequacy of the obtained results (Steps 1–4), in comparison with works [9–18].

Table 1.3 The value of reliability estimates

	n	m_{av}	k	t_{av}	s	$\Xi_{\text{Rete II}}$	Ξ_{Treat}	Ξ_{Leaps}	$\Xi_{\text{mod Rete II}}$
RB1	20	9	12	5	6	**0.7**	**0.68**	**0.77**	**0.89**
RB2	200	9	12	5	6	**0.76**	**0.67**	**0.75**	**0.85**
RB3	400	9	12	5	6	**0.65**	**0.67**	**0.77**	**0.88**
RB4	600	9	12	5	6	**0.66**	**0.69**	**0.8**	**0.87**
RB5	20	9	12	5	6	**0.69**	**0.7**	**0.76**	**0.87**
RB6	200	9	12	5	6	**0.68**	**0.71**	**0.72**	**0.88**
RB7	400	9	12	5	6	**0.69**	**0.67**	**0.74**	**0.89**
RB8	600	9	12	5	6	**0.7**	**0.66**	**0.77**	**0.95**
RB9	20	9	12	5	6	**0.66**	**0.7**	**0.75**	**0.92**
RB10	200	9	12	5	6	**0.67**	**0.72**	**0.78**	**0.93**
RB11	400	9	12	5	6	**0.64**	**0.71**	**0.73**	**0.97**
RB12	600	9	12	5	6	**0.67**	**0.73**	**0.76**	**0.98**
RB13	20	9	12	5	6	**0.6**	**0.69**	**0.74**	**0.92**
RB14	200	9	12	5	6	**0.74**	**0.73**	**0.75**	**0.94**
RB15	400	9	12	5	6	**0.69**	**0.7**	**0.76**	**0.96**
RB16	600	9	12	5	6	**0.7**	**0.69**	**0.78**	**0.97**

The disadvantages of the proposed approach method should include: increased computational complexity due to the construction of two types of models – object and relational, as well as the operation of two bio-inspired algorithms.

This method will allow:

– to carry out an assessment of the objects state with a given degree of reliability;

– to determine effective measures to increase the reliability of the assessment of the objects state while maintaining the specified efficiency.

The limitations of the research are the need to have an initial database on the objects state, the need to take into account the delay time for collection and proving information from intelligence sources.

The proposed method should be used to solve the problems of assessing the state of multidimensional objects in conditions of uncertainty and risks, which are characterized by high requirements for the reliability of the obtained data.

This research is a further development of researches aimed at developing method principles for increasing the efficiency of processing various types of data, which were published earlier [2, 4–6, 20].

The directions of further research should be aimed at reducing computing costs while processing various types of data in special purpose systems.

CONCLUSIONS

1. The procedures for implementing a method approach to assessing the state of complex hierarchical systems have been defined, thanks to additional and improved procedures that allow:

– to take into account the type of uncertainty and noise;

– to implement adaptive strategies for finding sources of ALA hunting;

– to determine the hunting strategy, taking into account the available computing resources of the system and the priority of the assessment;

– to take into account the available computing resources of the decision-making support system;

– to change the search area by individual ALA;

– to change the speed of the ALA in the specified search direction;

– to carry out the initial issuance of ALA, taking into account the type of uncertainty;

– to conduct a local and global search taking into account the degree of uncertainty and noisy data;

– to conduct training of knowledge bases, which is carried out by training the synaptic weights of the artificial neural network, the type and parameters of the membership function, as well as the architecture of individual elements and the architecture of the artificial neural network as a whole;

– to perform classification of ant nests according to the priority of assessment;

– to adjust the route of the ALA attack by ranking ant nests according to the level of ant pheromones;

– to avoid the problem of local extremum.

2. An example of the use of the proposed method approach has been provided, using the example of solving the task of determining the composition of an operational group of troops (forces) and elements of its operational construction. The specified example showed an increase in the effectiveness of the assessment of the state of hierarchical systems at the level of 22–25 % due to the use of additional improved procedures.

3. The procedures for the analysis and forecasting of the state of multidimensional objects using a metaheuristic algorithm are defined, which allows:

– to take into account available information on the state of multidimensional objects that determine awareness of their state;

– to implement various strategies for finding sources of nectar for BA;

– to determine the BA nectar extraction strategy taking into account the available computing resources of the system and the priority of the assessment;

– to change the search area by individual BA;

– to change the speed of BA movement in the specified search direction;

– to carry out the initial exposure of BA taking into account the type of uncertainty;

– to conduct a local and global search taking into account the degree of uncertainty;

– to conduct training of knowledge bases, which is carried out by training the synaptic weights of the artificial neural network, the type and parameters of the membership function, as well as the architecture of individual elements and the architecture of the artificial neural network as a whole;

– to classify nectar clusters according to the priority of assessment;

– to adjust the BA nectar collection route due to the ranking of nectar accumulations by the level stimulus;

– to avoid the problem of local extremum.

4. An example of the use of the proposed method was provided, on the example of solving the task of determining the composition of an operational group of troops (forces) and elements of its operational construction. The specified example showed an increase in the effectiveness of the operational efficiency of the state assessment of multidimensional objects at the level of 14–16 % due to the use of additional improved procedures.

5. The procedures of the method of increasing the reliability of the assessment of the object state have been determined, which allows:

– to increase the reliability of the assessment of the object state due to the parallel use of two bio-inspired algorithms;

– to take into account the degree of awareness of the object state due to the application of correction coefficients for the degree of awareness;

– to build both object and relational models, which allows to increase the reliability of the assessment of the objects state;

– to combine the results of the work of bio-inspired algorithms, which makes it possible to mutually verify the correctness of the work of each of the algorithms;

– to obtain the universal solution to the task of assessing the objects state with different degrees due to the hierarchical nature of their description;

– to carry out a simultaneous search for a solution in different directions.

6. An example of the use of the proposed method was provided on the example of solving the task of determining the composition of an operational group of troops (forces) and elements of its

operational construction. The specified example showed an increase in the reliability of the state assessment of objects on average by 20 % due to the use of additional improved 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

DOI: 10.15587/978-617-8360-13-9.CH2

Oleksandr Gaman, Svitlana Kashkevich,
Oleh Shknai, Andrii Radchenko, Serhii Neronov, Zaur Musaiev

CHAPTER 2

METHODS OF PROCESSING VARIOUS DATA IN INTELLIGENT SYSTEMS FOR MANAGEMENT OF THE NETWORK AND SERVER ARCHITECTURE OF THE INTERNET OF COMBAT THINGS

ABSTRACT

In this chapter of the research, a method of processing various types of data in intelligent management systems of the network and server architecture of the internet of military equipment is proposed.

The basis of this research is the theory of artificial intelligence, namely evolving artificial neural networks, basic genetic algorithm procedures, neuro-fuzzy expert systems, as well as bio-inspired algorithms.

In the course of the research, the authors proposed:

– a complex model of processing various types of data in intelligent decision-making support systems;

– a method of processing various types of data in intelligent management systems of network and server architecture;

– a method of increasing the efficiency of processing various types of data in intelligent management systems of network and server architecture.

The use of methods of processing various types of data in intelligent management systems of the network and server architecture of the internet of military equipment:

– to reduce the probability of premature convergence of the metaheuristic algorithm while processing various types of data in intelligent management systems of the network and server architecture of the internet of military equipment;

– to maintain a balance between the speed of convergence of the metaheuristic algorithm and diversification while processing various types of data in the intelligent control systems of the network and server architecture of the internet of military equipment;

– to take into account the type of uncertainty and noise of the data of the metaheuristic algorithm while processing various types of data in the intelligent control systems of the network and server architecture of the internet of military equipment;

– to take into account the available computing resources of the system while processing various types of data in the intelligent management systems of the network and server architecture of the internet of military equipment;

– to take into account the priority of search by swarm agents of the meta-heuristic algorithm while processing various types of data in intelligent management systems of the network and server architecture of the internet of military equipment;

– to conduct an initial display of flock individuals taking into account the type of uncertainty;

– to conduct accurate training of metaheuristic algorithms;

– to conduct a local and global search taking into account the degree of data noise while processing various types of data in intelligent management systems of the network and server architecture of the internet of military equipment.

KEYWORDS

Internet of military equipment, processing of various types of data, combined systems, reliability and efficiency.

2.1 THE DEVELOPMENT OF A COMPLEX MODEL FOR PROCESSING VARIOUS DATA

The development of IoT and its adaptation on a large scale will require both new theoretical research and significant funding. The distribution of network resources of various technical means will require new approaches and intelligent automation. An important issue will be the perception of the amount of information that IoT will provide.

The most promising option for increasing the efficiency (and, as a consequence, the efficiency) of information processing is the use of approaches based on artificial intelligence. One such toolkit of artificial intelligence is the expert system.

Currently, expert systems have become the main tool used to solve various types of tasks (interpretation, forecast, diagnosis, planning, design, control, debugging, instruction and management) in a wide variety of problem areas [1–9]. The functioning of expert systems is based on the knowledge model [10–18]. It contains a set of principles that describe the state and behavior of the research object. The most widely used knowledge model for expert systems is the production model due to its simplicity, ease of processing and comprehensibility of the end user [19–32].

However, recently, fuzzy expert systems have become widespread. This type of expert systems is based on a set of rules that use linguistic variables and fuzzy relations to describe the state and behavior of the object under investigation [33–54].

The rules presented in this form are the closest to natural language, so there is no need to use a separate expert knowledge engineer to create and edit the rules. In most cases, they can be edited by the expert itself practically without special training [55–67].

One of the most important problems inherent in knowledge-based systems is the problem of knowledge representation [65–71]. This is explained by the fact that the form of knowledge representation significantly affects the characteristics and properties of the system. In order to operate all kinds of knowledge from the real world with the help of a personal computer, it is necessary to carry out their simulation.

In such cases, it is necessary to distinguish knowledge intended for processing by computing devices from knowledge used by humans. In addition, with a large amount of knowledge, it is desirable to simplify the sequential management of individual elements of knowledge.

Taking into account the above, the aim of the research is to develop a complex model of processing various types of data in intelligent decision-making support systems. The object of research is the intelligent decision-making support system. The subject of research is the knowledge representation model in intelligent decision-making support systems.

On the basis of the decision-making support system model [6], as well as the model of a complex task, which is the processing of various types of data in intelligent server and network architecture management systems, let's build a model of intelligent DMSS:

$$DMSS = <PRT, prt^{dm}, R^{dm}, M^{dm}>, \quad (2.1)$$

where $PRT = \{prt_q \mid q = 1,\ldots,N_{prt}\}$ is a set of expert models; prt^{dm} is a model of a decision maker; $R^{dm} = \{r^{dmq} \mid q = 1,\ldots,N_{prt}\}$ is the relationship between the decision maker and experts, for example, the relationship of information exchange; M^{dm} are the methods of processing information received from experts.

Each expert works strictly in its field of expertise $S_{prtq} \in S$, where S is the set of all fields of knowledge necessary to solve the task of processing various types of information and does not deal with any subtasks other than its own, $S_q \cap S_w = \varnothing$, with $q, w = 1,\ldots,N_{prt}$; $q \neq w$.

Based on considerations from research [5] and taking into account that in real tasks subtasks are solved by experts step by step, the expert model can be presented:

$$prt_q = <B_{prof}, B_{theor}, B_{prec}, B_{facts}, MET_{prtq}, S_{prtq}, In_{prtq}, \Delta t>, \quad (2.2)$$

where B_{prof} is the production base of professional knowledge; B_{theor} is the production base of theoretical knowledge; B_{prec} is the basis of precedents (experience); B_{facts} is the basis of facts; MET_{prtq} is a set of reasoning methods; S_{prtq} is the description of the expert's field of knowledge; In_{prtq} is an interpreter that ensures the execution of a sequence of rules for solving a task based on facts and rules stored in databases and knowledge; Δt is the period of issuing intermediate decisions by experts.

The model of the decision maker can be built by analogy with (2.2):

$$prt^{dm} = <B_{prof}, B_{theor}, B_{prec}, B_{facts}, B_{ext}, MET_{prldm}, S_{prldm}, In_{prldm}, E, T>, \quad (2.3)$$

where B_{ext} is the production knowledge base on how to perform reduction, aggregation, comparison and coordination; E is a set of coordinating processes; T is the processing time of various types of data.

The expression (2.3) in comparison with expression (2.2) has significant differences. Production knowledge base B_{ext} about how the decision maker manages the processing of various types of data. This knowledge is obtained from other experts.

The set E describes how the decision maker can coordinate the work of experts, B_{prof} is base of professional knowledge, B_{theor} is the basis of theoretical knowledge, B_{prec} is the basis of precedents (experience).

Let's consider how the intelligent DMSS functions according to the work (2.1). Let the decision maker be given a task prb^u, which is reduced to subtasks $prb_1^h,\ldots,prb_{N_h}^h$. Analyzing the expression (2.1) and (2.3), as well as relying on the practice of processing various types of data, it is possible to draw the following conclusions: GL^h contained in B_{pred} and B_{facts} is the experience combined with facts allows the expert to determine what result should be obtained; MET^h contained in $B_{prof}, B_{theor}, B_{pred}, MET_{prti}, S_{prti}, In_{prti}$; DAT^h contained in B_{facts}.

In traditional DMSS, described, for example, in work [5], each expert prt_q, $q=1,\ldots,N_{prt}$, having received its subtask of processing various types of data prb_j^h, $j=1,\ldots,N_h$, finds its solution using its professional skills B_{prof} and theoretical knowledge B_{theor}.

After finishing the process of processing various types of data, it issues a result $sol^{hj} \in SOL_j^h$, where SOL_j^h is a set of results of solving the task prb_j^h, which can be written as a correspondence ψ_4:

$$\psi_4:\ DAT^h \otimes B^U \to SOL^h,\ B^U = B_{prof} \cup B_{theor}. \tag{2.4}$$

Compliance elements ψ_4 are the tuples $\left(\left(\left\{dat_\sigma^h\right\},\{b_\beta^u\}\right), sol_\gamma^h\right)$, with $\sigma=1,\ldots,N_{dath}$; $\beta=1,\ldots,N_b$; $\gamma=1,\ldots,N_{sh}$, where the first component is a two-component vector consisting of a list of initial data $\left\{dat_\sigma^h\right\}$, $dat_\sigma^h \in DAT^h$ and list of knowledge $\{b_\beta^u\}$, $b_\beta^u \in B^U$ (professional knowledge are production rules; theoretical knowledge are analytical dependencies) and the second is the result $sol_\gamma^h \in SOL^h$ processing of various types of data prb^h.

Conformity ψ_4 is not a function (cannot be written analytically or calculated by numerical methods) because the knowledge of an expert and the results of processing an element of heterogeneous data can be represented in natural language. It is ambiguous, because with an incomplete set of initial data (a priori uncertainty), the expert can offer several options for results, subjectively, because each decision on the processing of different types of data prb^h corresponds to at least one element of $DAT^h \otimes B^U$ and not injunctive, because not every element of $DAT^h \otimes B^U$ corresponds to the solution of the task prb^h.

Let's indicate the number of stages into which experts divide the process of solving partial tasks N_{sol} and sol_l^h is the result of solving a partial task at the l-th stage, $l=1,\ldots,N_{sol}$. A time interval is allocated to the stage of processing various types of data Δt. Because in practical tasks, the total time T for solving the task of processing various types of data prb^u, strictly limited and time Δt between refinements of the task of processing various types of data is constant, the number of stages is determined by the formula:

$$N_{sol} = T/\Delta t. \tag{2.5}$$

It should be noted that in the process of solving a partial task of processing various types of data prb^h, through the coordinating influences of the decision maker, the original data DAT^h in expression (2.4) can be modified – additional information is entered or outdated information is replaced with new information.

Let DAT_l^h output data for the l-th stage, $l=1,\ldots,N_{sol}$. Then DAT_1^h is the initial data received from the decision maker, moreover $DAT_1^h = DAT^h$ and DAT_l^h, $l=2,\ldots,N_{sol}$ is the output data of the following stages.

The index l means the number of the stage in which the raw data is used. Let's define DAT_{l+1}^h as initial data of the $l+1$-th stage, obtained after the coordinating influences of the decision maker regarding the change of the data of the l-th stage.

The scheme of the sequence of stages of the expert's work in finding a solution to a partial task π^h can be expressed as follows:

$$DAT_l^h \otimes B^U \otimes \left\{sol_1^h\right\}\otimes,\ldots,\otimes\left\{sol_{l-1}^h\right\} \Rightarrow \left\{sol_1^h\right\},\ l=1,\ldots,N_{sol}. \tag{2.6}$$

Output data DAT_l^h, $l=1,\ldots,N_{sol}$ at each stage are supplemented by coordinating influences $e^{\alpha}\in E$, issued by the decision maker to the expert, which are determined on the basis of the integrated result of solving the task of processing various types of data prb^u on $l-1$-th stage. Let's assume that the expert is given one coordinating influence of one type. Let's determine the correspondence ψ_5:

$$\psi_5:\ \left\{sol_l^u\right\}\otimes B_{ext} \to E,\ l=1,\ldots,N_{sol}-1. \tag{2.7}$$

The maximum value of l is equal to $N_{sol}-1$ because after N_{sol} stage, it is no longer possible to use coordination, since the final result is obtained.

sol_l^u is the integrated result of solving the task of processing various types of data prb^u at the l-th stage; $E=\{\varepsilon_1,\ldots,\varepsilon_{N_\varepsilon}\}$ is a set of type vectors $\left(e_1^1,\ldots,e_{N_{prt}}^6\right)$, each component of which is a coordinating action for the expert, $e_q^{\alpha}\in E, q=1,\ldots,N_{prt}$.

Since the knowledge of integration is included B_{ext} the decision maker (4) is the integrated result sol_l^u solving the complex task of processing various types of data prb^u can be written like this:

$$\left\{sol_l^{h1}\right\}\otimes\ldots\otimes\left\{sol_l^{hN_h}\right\}\otimes B_{ext} \to \left\{sol_l^u\right\}, \tag{2.8}$$

where $sol_l^{h1},\ldots,sol_l^{hN_h}$ are the solving partial tasks for processing various types of data $prb_1^h,\ldots,prb_{N_h}^h$ in accordance.

Compliance elements ψ_5 are the tuples $\left(\left(sol_l^u,\{b_{ext}^{\mu}\}\right),\varepsilon_p\right)$, with $l=1,\ldots,N_{sol}$, $\mu=1,\ldots,\mu_{N_\mu}$, $p=1,\ldots,N_\varepsilon$, where the first component is a two-component vector consisting of the integrated result sol_l^u solving the task of processing various types of data prb^u at the l-th stage and the list of knowledge used by the decision maker on how to perform the comparison

$\{b_{ext}^{\mu}\}$, $b_{ext}^{\mu} \in B_{ext}$ and the second component is a vector whose elements are coordinating actions $e \in E$ for an expert.

On N_{sol}-th stage ($l = N_{sol}$) vector of coordinating influences $\varepsilon = \left(e_1^{\alpha},\ldots,e_{N_{prt}}^{\alpha}\right)$, $\alpha = 6$, thus, the decision maker does not issue coordinating influences to experts, but only aggregates (integrates solutions to partial tasks prt^h into a single, integrated solution sol_l^u of a difficult task prb^u) the results of their work.

If the integrated result is obtained sol_l^u and does not suit the decision maker, it must revise the original tasks prb^u, so change DAT_l^h for prb^h or change the list of your knowledge B_{ext} and expert knowledge B_{prof} and after that, initiate the re-operation of the intelligent DMSS.

In accordance, ψ_5 is not a function (cannot be written analytically and calculated), since the knowledge of the decision maker and the integrated result of solving the task prb^u can be represented in natural language. It is unambiguous, as each expert is assigned a specific coordinating action e_q^{α} and therefore compliance ψ_5 uniquely defines only one vector $\varepsilon \in E$. It is subjective, because for each vector $\varepsilon \in E$ at least one element matches $\{sol_l^u\} \otimes B_{ext}$ and not inductively, because not to every element $\{sol_l^u\} \otimes B_{ext}$ corresponds to a vector $\varepsilon \in E$.

The choice of whether the intermediate result of the solution of a partial task of processing various types of data prb_w^h, $w = 1,\ldots,N_h$ is ready and expert $prt_q \in PRT$, $q = 1,\ldots,N_{prt}$ accepts based on experience B_{prec} (2.3).

The functional multi-agent system model presented in work [4] is adopted as the basic model for processing homogeneous data in intelligent DMSS. The choice of this model is explained by the fact that the complexity of the homogeneous tasks to be solved is not great – there are relatively few input and output parameters and expert rules, so there is no need to investigate the micro-level of subtasks.

It can also be noted that the decomposition of the task of processing heterogeneous data into a set of tasks of processing homogeneous data is transparent, it is not difficult to establish a connection between subtasks and elements of the intelligent DMSS.

The conceptual model of an element of a multi-agent intelligent DMSS for processing homogeneous data can be presented:

$$res^e = R_1^{resmet}\left(res^e, met^e\right) \circ R_1^{respr}\left(res^e, pr^{ei}\right) \circ R_1^{respr}\left(res^e, pr^{eo}\right) \circ R_1^{resst}\left(res^e, st^e\right) \circ$$
$$\circ R_1^{stst}\left(st^e(t), st^e(t+1)\right) \circ R_1^{prst}\left(pr^{ei}(t), st^e(t+1)\right) \circ R_1^{stpr}\left(st^e(t), pr^{eo}(t)\right), \tag{2.9}$$

where R_1^{stst}, R_1^{prst}, R_1^{stpr} is the relationship "state – state", "input – state", "state – output", respectively.

Among the crowd $MET^e = \{met_y^e \mid y = 1,\ldots,N_{met}\}$ autonomous methods will be highlighted met_1^e are analytical calculations, met_2^e are neurocomputing, met_3^e are fuzzy calculations, met_4^e is reasoning based on experience; met_5^e are evolutionary calculations, met_6^e are statistical calculations, met_7^e is logical reasoning. If between element res^e and autonomous method met_y^e relationship is established $R_1^{resmet}\left(res^e, met_y^e\right)$, let's denote the element res^{ey}. The appropriate calculation method is selected according to expression (2.1).

Relations R_1^{prpr}, R_2^{prpr} (2.9) are given on sets of variables DAT^u, GL^u and sets of variables DAT^h, GL^h from the processing of homogeneous data included in the task of processing heterogeneous data.

The following cases are possible:

1) a set of variables for prb^u coincides with the set of variables for prb^h, so $DAT^u = DAT^h$, $GL^u = GL^h$;

2) a set of variables for prb^h is a subset of the corresponding set u_{prb}, so $DAT^h \subset DAT^u$, $GL^h \subset GL^u$;

3) a set of variables of a subset of the corresponding set prb^h, so $DAT^u \subset DAT^h$, $GL^u \subset GL^h$. The second variant is typical for the analyzed task of processing heterogeneous data, because during the solution of the task of processing heterogeneous data, the initial data is divided between different partial tasks of processing homogeneous data.

The extension of model (2.9) is performed based on the following considerations. In the process of coordination, the intermediate states of the solution of partial tasks for the processing of homogeneous data are controlled.

In the accepted notation (2.9), these states mean the states (decision results) of functional elements res^e, simulating the solutions of partial tasks for processing homogeneous data prb^h. From the analysis of these states, the "input" properties change during coordination pr^{ei} of one or more elements res^e.

To take this fact into account, let's introduce a triple into the conceptual model (2.9). $R_1^{stpr}\left(st^u(t), pr^{ui}(t+1)\right)$. In other words, based on the state of intellectual DMSS $st^u(t)$ at time t, the output data changes $pr^{ui}(t+1)$ for intelligent DMSS, but already at the moment of time $t+1$, so for the next iteration.

Plural R_1^{stpr} establishes relationships between state $st^u(t)$ of hybrid res_A^u (2.9) at the moment of the model time t and the state of the inputs of one or more elements res^e in the next step. To make the necessary change of inputs pr^{ei} of one or more functional elements res^e (2.9), let's enter a triple $R_1^{stact}\left(st^u, act^{ek}\right)$, where $ACT^{ek} = \left\{act_1^{ek\alpha}, \ldots, act_{N_{prt}}^{ek\alpha}\right\}$ is the set of concepts denoting coordinating actions, which is identical to the set of coordinating actions E, where α is the type of coordinating action, $\alpha = 1,\ldots,6$. In the coordination algorithm, the coordinating actions are described by the knowledge base B_{ext}. Plural R_1^{stact} is a set of relations between states st^u of intellectual support system res_A^u at the moment of the model time t and the necessary coordinating actions ACT^{ek}.

The modified conceptual model of the intelligent DMSS for processing homogeneous data with coordination has the following form:

$$res_A^u = res_A^u \circ R_1^{stpr}\left(st^u(t), pr^{ui}(t+1)\right), \tag{2.10}$$

and a modified model of an element of an intelligent DMSS for processing homogeneous data:

$$res^e = res^e \circ R_1^{stact}\left(st^u, act^{ek}\right). \tag{2.11}$$

Relations R_1^{stpr} and R_1^{stact} are not set in advance, like R_1^{stst}, R_1^{prst}, R_1^{stpr} are fixed during the operation of the intelligent DMSS for processing homogeneous data and are the result of the solution of the k-task prb^k.

The following is a conceptual model of the operation of an intelligent DMSS for processing homogeneous data, built according to the expressions (2.10), (2.11):

$$st^{ek}(t_0') \Rightarrow \begin{Bmatrix} st^{e1}(t_0) \\ st^{e2}(t_0) \end{Bmatrix} \to \begin{Bmatrix} st^{e1}(t_1) \\ st^{e2}(t_1) \end{Bmatrix} \Rightarrow st^{ek}(t_0') \to st^{ek}(t_1') \Rightarrow$$
$$\Rightarrow \begin{Bmatrix} st^{e1}(t_1) \\ st^{e2}(t_1) \end{Bmatrix} \to \begin{Bmatrix} st^{e1}(t_2) \\ st^{e2}(t_2) \end{Bmatrix} \Rightarrow st^{ek}(t_1') \to st^{ek}(t_2') \Rightarrow \ldots \Rightarrow$$
$$\Rightarrow \begin{Bmatrix} st^{e1}(t_{p-1}) \\ st^{e2}(t_{p-1}) \end{Bmatrix} \to \begin{Bmatrix} st^{e1}(t_p) \\ st^{e2}(t_p) \end{Bmatrix} \Rightarrow st^{ek}(t_{p-1}') \to st^{ek}(t_p'), \qquad (2.12)$$

where "$\Rightarrow$" is the relationship R^{stst}, which connect the states from different subspaces and specify the transition from one homogeneous space to others during the functioning of the intelligent DMSS for processing homogeneous data; "$\to$" is the transition between states within the corresponding subspace.

Transitions "$\Rightarrow$" from the element subspace res_k^{e7} model the issue of coordinating influences from the decision maker to experts while processing homogeneous data. And the set of transitions "$\Rightarrow$" and "$\to$" allows to simulate and trace the process of self-organization in the process of the operation of an intelligent DMSS for the processing of homogeneous data.

In the expression (2.12), curly brackets indicate the beginning and end of the parallel work of the functional elements of the intelligent DMSS for processing homogeneous data. It can be seen from the model that after each fixation " $\}\Rightarrow$" of states, functional elements res_q^{ey} control of the element res_k^{e7} is transferred and after it changes its state, control is transferred to a group of functional elements.

This model is related to the conceptual model:

$$\begin{Bmatrix} st^{e1}(t) \to st^{e1}(t+1) \to \ldots \to st^{e1}(t+n) \\ st^{e2}(t) \to st^{e2}(t+1) \to \ldots \to st^{e2}(t+n) \end{Bmatrix}. \qquad (2.13)$$

The simulation of the proposed model was carried out in the MathCad 2014 software environment for evaluating the radioelectronic environment.

The initial setting of the membership functions of the set of terms of the neuron fuzzy expert system has been performed, since all sources of radio radiation have different characteristics. The experts indicated which values of primary and calculated parameters are considered high for radio emission devices, which are average and which are low. The membership functions for the analysis of the radio-electronic situation are presented in the specified form according to the formula:

1) (PJ="H") and (KOV="H") and (UN="H") and (PW="L") $\rightarrow$ *(BER="H"),*

...

81) (PJ="L") and (KOV="L") and (UN="L") and (PW="H") $\rightarrow$ *(BER="L"),*

82) (FJ="L") and (FW="L") and (UN="H") and (PW="H") $\rightarrow$ *(BER="N"),*

...

108) (SC="L") and (KOV="L") and (UN="H") and (PJ="L") $\rightarrow$ *(BER="N"),*

In this example, part of the rule base of the neuro-fuzzy expert system is given. In the main base of rules there are rules not only with connections of conditions with the help of *T*-norms, but also with the help of *T*-conorms and with negations of conditions.

In the worst case, to find a solution, the system should check all the rules contained in the rule base. Thus, it is necessary to check 405 conditions and calculate 297 *T*-norm operations. This is an unacceptably long process, given the limitations of the hardware.

The input data for the neuro-fuzzy expert system are indicators of the power of transmitters of radio-emitting devices, the type of signal-code structures, the uncertainty of the radio-electronic situation, the frequency of radiation of radio-emitting devices. After passing through the phase of fuzzification, the system received fuzzy estimates for each monitored parameter.

The results of the assessment of the state of the radio-electronic situation for various presentation models are shown in **Fig. 2.1**.

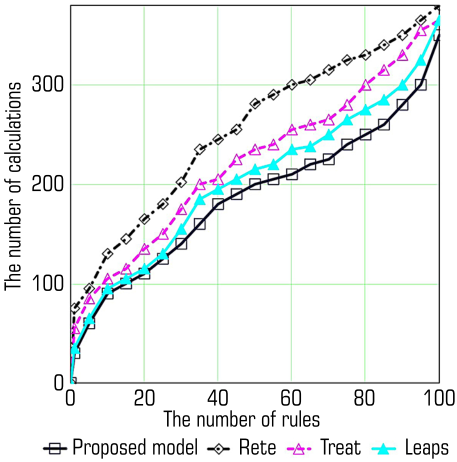

Fig. 2.1 Comparison of the efficiency of the obtained estimate for different models

Features of radio-electronic situation analysis systems are: a large number of analyzed parameters; dynamic change of the electronic environment; functioning in conditions of uncertainty about the state of the radio-electronic situation; constant updating of the signal database; functioning under the influence of natural and intentional disturbances.

As a result of the conducted research, the following was established:

– a complex task of processing heterogeneous data under certain restrictions and assumptions can be divided (performed decomposition) into a number of separate partial tasks of processing homogeneous data;

– taking into account the specifics of the data that must be processed in intelligent DMSS, the availability of computing power, it is necessary to carefully approach the use of mathematical apparatus for modeling and processing of data circulating in the specified systems;

– with homogeneous data, it is advisable to use the model described by expressions (9)–(12) and after adding new types of data, it is advisable to increase the models of the lower level.

The direction of further research should be the improvement of information processing methods in intelligent decision-making support systems.

2.2 THE METHOD OF PROCESSING VARIOUS TYPES OF DATA IN INTELLIGENT MANAGEMENT SYSTEMS OF NETWORK AND SERVER ARCHITECTURE

This method is based on the duck flocking algorithm (DFA). There are two rules that must be followed in the foraging process of duck agents (DA). Rule One: While searching for food, DA with strong search ability are positioned closer to the center of the food source, which attracts other DA to move closer to them. The updated location is also affected by nearby DA. Rule two: while looking for food, all DA approach food. The next position is affected by the adjacent DA and the food position or the DA leader. The method of processing various types of data in intelligent management systems of network and server architecture consists of the following sequence of actions:

Step 1. Input of initial data. At this stage, the raw data available on the intelligent network and server architecture management system are entered.

The main parameters of DFA are the maximum number of iterations $Iter_{max}$, the population size NP, the number of decision variables n, the probability of the presence of a predator Pdp, the scaling factor sf, the migration constant G_c, as well as the upper and lower bounds for the decision variable FS_U and FS_L:

$$FS_{i,j} = \left(FS_L + \left(rand(\ )\right) * \iota\right) * \left(FS_U - FS_L\right), \tag{2.14}$$

$$i = 1, 2, \ldots, NP,\ j = 1, 2, \ldots, n,$$

where $rand()$ is a uniformly distributed random number in the range [0, 1]; ι is a correction coefficient that determines the degree of a priori information about the state intelligent network and server architecture management system.

Step 2. Numbering of DA in the flock, i, $i \in [0, S]$. At this stage, each DA is assigned a serial number.

Let's add an identifier for each duck in the Duck structure so that each agent has its own unique number.

Step 3. Determination of the initial speed of DA. The initial speed v_0 of each DA is determined by the following expression:

$$v_i = \left(v_1, v_2, \ldots, v_S\right), v_i = v_0. \tag{2.15}$$

Step 4. Exhibition of DA on the search plane.

At this stage, the DA is issued taking into account the type of uncertainty about the state intelligent network and server architecture management system, and the basic model of its state is initialized [2].

Let's suppose that the expression for a randomly generated initial position in a D-dimensional search space is:

$$X_i = L_b + \left(U_b - L_b\right) \cdot o, \tag{2.16}$$

where X_i is the spatial position of the i-th DA (i=1, 2, 3, ..., N) in the DA group; N is the population size; L_b and U_b represent the upper and lower bounds of the search space, respectively; o is a matrix of random numbers between (0, 1).

Step 5. Search for food sources by individual DA individuals.

After taking turns, the DA flock arrived at a location with more food. Each DA gradually disperses and begins to search for food, this process is defined as follows:

$$X_i^{t+1} = \begin{cases} X_i^t + \mu \cdot X_i^t \cdot sign\left(r - 0.5\right), P > rand; \\ X_i^t + CF_1 \cdot \left(X_{leader}^t - X_i^t\right) + CF_2 \cdot \left(X_j^t - X_i^t\right), P \le rand, \end{cases} \tag{2.17}$$

where $sign\left(r - 0.5\right)$ affects the process of finding food and it can be set to –1 or 1; μ is the global search control parameter; P is the probability of conversion of the search of the reconnaissance phase; CF_1 and CF_2 are the coefficients of cooperation and competition between DA at the search stage, respectively; X_{leader}^t is the best position of the DA of the current historical value in the t-th iteration; X_j^t is the DA around X_i^t in the search for feeding the DA group in the t-th iteration.

Parameter μ can be calculated as follows:

$$\mu = K \cdot \left(1 - t/t_{\max}\right), \tag{2.18}$$

where K is calculated by the formula:

$$K = \sin\left(2 \cdot rand\right) + 1. \tag{2.19}$$

The search range of a flock of ducks is wider when $\mu > 1$. This non-linear strategy is used to improve the global search capability of the proposed gas station. In addition, this phase can also integrate two parameters C_1 and C_2 will be factored into the update formula used to balance DA positions in the research phase.

The update formula is described as follows:

$$X_i^{t+1} = X_i^t + C_1 \cdot \mu \cdot \left(X_i^t - C_2 \cdot X_{leader}^t \right) \cdot sign\left(r - 0.5 \right) +$$
$$+ CF_1 \cdot \left(X_{leader}^t - X_i^t \right) + CF_2 \cdot \left(X_j^t - X_i^t \right), \tag{2.20}$$

where μ is the control parameter of the global DA search; C_1 and C_2 are the parameters for adjusting the balance of the position; C_1 is a random number in (0, 1). C_2 can be set not only as random, but also as a constant, for example 0, 1 or others; CF_1 and CF_2 are the coefficients of cooperation and competition between DA at the search stage; X_{leader}^t is the best position of the DA of the current historical value in the t-th iteration; X_j^t is the number of individuals around X_i^t in the search for food by the DA group in the t-th iteration.

Step 6. Classification of food sources for DA.

Locations of the best DA food source (minimum fitness) are considered watercress (FS_{ht}), locations from the following three food sources have vegetables (FS_{at}) and the rest are considered common plants (FS_{nt}):

$$FS_{ht} = FS(\text{sorte\_index}(1)), \tag{2.21}$$

$$FS_{at}(1:3) = FS(\text{sorte\_index}(2:4)), \tag{2.22}$$

$$FS_{nt}(1:NP\text{-}4) = FS(\text{sorte\_index}(5:NP)). \tag{2.23}$$

Step 7. Sorting of the best DA individuals. The selection of the best individuals is carried out using the improved genetic algorithm proposed in work [13]. While searching for food, the strongest DA, which is the largest in the flock, directs the other DA in the group to look for food. The weight in DA is similar to the quality of the objective function values of the candidate solutions. Therefore, the best solution variant with the best value for the objective function is considered to be the largest DA in the group. This search behavior of DA leads to different scanning areas of the search space, which improves the exploration ability of DA in global search.

Step 8. The search for food sources by the DA flock.

After that, a flock of ducks collects enough food to meet its own nutritional needs. This process is closely related to the suitability of the position of each DA and is defined as follows:

$$X_i^{t+1} = \begin{cases} X_i^t + \mu \cdot \left(X_{leader}^t - X_i^t \right), f\left(X_i^t \right) > f\left(X_i^{t+1} \right); \\ X_i^t + KF_1 \cdot \left(X_{leader}^t - X_i^t \right) + KF_2 \cdot \left(X_k^t - X_j^t \right), \text{otherwise}, \end{cases} \tag{2.24}$$

where μ is the controlling parameter of the global search at the stage of the group food search of DA; KF_1 and KF_2 are the coefficients of cooperation and competition between DA at the stage of group foraging of DA; X^t_{leader} is the best position of the DA of the current historical value in the t-th iteration; X^t_k and X^t_j are the numbers of DA around X^t_i in the search for food by the DA group in the t-th iteration, where $k \neq j$.

All parameter values CF_1, CF_2, KF_1 and KF_2 are in the range of values (0, 2), and the calculation formula can be summarized as follows:

$$CF_i \text{ or } KF_i \leftarrow \frac{1}{FP} \cdot rand(0,1)\ (i=1,2), \tag{2.25}$$

where FP is a constant, it is set to 0.618; *rand* is a random number in works (0, 1).

Step 9. Checking the presence of a predator. At this stage, AM is checked for the presence of predators. If there are predators, go to Step 8. If there are no predators, the end of calculations.

End of the algorithm.

The software implementation of the method of processing various types of data in intelligent management systems of network and server architecture is given below:

```
// Duck Swarm Algorithm.cpp
#include <iostream>
#include <vector>
#include <cmath>
#include <limits>
#include <cstdlib>
#include <ctime>
using namespace std;
const int POPULATION_SIZE = 50;
const int DIMENSIONS = 2;
const int MAX_ITERATIONS = 1000;
const double INERTIA = 0.5;
const double LEARNING_RATE = 2.0;
struct Duck {
   vector<double> position;
   vector<double> velocity;
   double fitness;
   Duck(int dimensions) {
      position.resize(dimensions);
      velocity.resize(dimensions);
      fitness = numeric_limits<double>::infinity();}};
double fitnessFunction(const vector<double>& position) {
```


CHAPTER 2

```
    double sum = 0.0;
    for (double x : position) {
        sum += x * x;}
    return sum;}
void updatePosition(Duck& duck, const vector<double>& bestPosition) {
    for (int i = 0; i < DIMENSIONS; ++i) {
        duck.velocity[i] = INERTIA * duck.velocity[i] +
            LEARNING_RATE * ((double)rand() / RAND_MAX) * (bestPosition[i] – duck.position[i]);
        duck.position[i] += duck.velocity[i]; }}
void duckSwarmOptimization() {
    vector<Duck> population(POPULATION_SIZE, Duck(DIMENSIONS));
    vector<double> globalBestPosition(DIMENSIONS);
    double globalBestFitness = numeric_limits<double>::infinity();
    for (Duck& duck : population) {
        for (int i = 0; i < DIMENSIONS; ++i) {
            duck.position[i] = ((double)rand() / RAND_MAX) * 10 – 5;
            duck.velocity[i] = ((double)rand() / RAND_MAX) * 2 – 1;  }
        duck.fitness = fitnessFunction(duck.position);
        if (duck.fitness < globalBestFitness) {
            globalBestFitness = duck.fitness;
            globalBestPosition = duck.position;
        } }
    for (int iter = 0; iter < MAX_ITERATIONS; ++iter) {
        for (Duck& duck : population) {
            updatePosition(duck, globalBestPosition);
            duck.fitness = fitnessFunction(duck.position);
            if (duck.fitness < globalBestFitness) {
                globalBestFitness = duck.fitness;
                globalBestPosition = duck.position; }}
      cout << "Iteration " << iter + 1 << " Best decision: " << globalBestFitness << endl;}
    cout << " Optimized solution: ";
    for (double x : globalBestPosition) {
        cout << x << " "; }
    cout << endl;}
int main() {
    srand(time(0));
    duckSwarmOptimization();
    return 0;
}
```

2.3 THE METHOD OF INCREASING THE EFFICIENCY OF PROCESSING VARIOUS TYPES OF DATA IN INTELLIGENT MANAGEMENT SYSTEMS OF NETWORK AND SERVER ARCHITECTURE

In this chapter of the dissertation research, an optimizer based on the simulation of reptile behavior (the case of crocodiles and alligators) is proposed – a population-based stochastic algorithm that uses reptilian agents (RA) as search agents.

The method of increasing the efficiency of processing various types of data in intelligent management systems of network and server architecture consists of the following sequence of actions:

Step 1. Input of initial data. At this stage, the main parameters of the algorithm are determined, such as:

– the type of task being solved;

– the number of agents in the population;

– the type of data (structured, unstructured), archived, real-time data;

– the number of variables characterizing the task being solved;

– available computing resources of the system;

– the type of uncertainty about the hierarchical system (complete uncertainty, partial uncertainty, complete awareness);

– the volume and type of the research sample;

– the volume and type of test sample;

– artificial neural network architecture, etc.

Step 2. Creation of a RA flock. Initialization of the RA population X_i ($i=1, 2, ..., n$) takes place. RA set form a population described by the matrix X. The initial population of RA in this set of algorithms is generated taking into account the uncertainty about the state of the intelligent management system of the network and server architecture based on the constraints of the problem under consideration. Members of the RA population are search agents in the solution space, providing candidate values for the problem variables based on their positions in the search space. Mathematically, each member of the general population is a vector, the number of elements of which is equal to the number of task variables.

The issuance of RA is carried out taking into account uncertainty about the data circulating in the system based on the basic system model and circulating data models [2, 19, 21]:

$$X=\begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m}=\begin{bmatrix} x_{1,1}\times\iota_{1,1} & \cdots & x_{1,d}\times\iota_{1,d} & \cdots & x_{1,m}\times\iota_{1,m} \\ \vdots & \ddots & \vdots & & \vdots \\ x_{i,1}\times\iota_{i,1} & \cdots & x_{i,d}\times\iota_{i,d} & \cdots & x_{i,m}\times\iota_{i,m} \\ \vdots & & \vdots & \ddots & \vdots \\ x_{N,1}\times\iota_{N,1} & \cdots & x_{N,d}\times\iota_{N,d} & \cdots & x_{N,m}\times\iota_{N,m} \end{bmatrix}_{N\times m}, \tag{2.26}$$

where X is the population matrix of RA; X_i is the i-th member of the RA flock (solution candidate); $x_{i,d}$ is the d-th dimension in the search space (decision variable); N is the number of RA; m is the number of decision variables.

Step 3. Numbering of RA in the flock, i, $i\in[0, S]$. At this stage, each RA is assigned a serial number. This makes it possible to determine the parameters of finding a solution for each individual in the flock.

Step 4. Determination of the initial speed of the RA.

Initial speed v_0 of each RA is determined by the following expression:

$$v_i = \left(v_1, v_2, \ldots, v_S\right), v_i = v_0. \tag{2.27}$$

The process of updating the RA population is based on the simulation of two strategies: the exploration phase and the exploitation phase.

Step 5. Preliminary assessment of the RA search area. In this procedure, the natural language search section is determined precisely by the halo of RA existence. Given that food sources for RA are food of animal origin, it is advisable to sort the suitability of food sources (Step 6).

Step 6. Classification of food sources for RA.

The location of the best food source (minimum fitness) is considered to be (FS_{ht}) the animal food (carrion) that is nearby and requires the least energy expenditure to find and obtain it. Delicacy food of animal origin will be denoted as FS_{at}.

Other non-priority food sources (food that is necessary for the survival of individuals) will be designated as FS_{nt}:

$$FS_{ht}=FS(\text{sorte\_index}(1)), \tag{2.28}$$

$$FS_{at}(1:3)=FS(\text{sorte\_index}(2:4)), \tag{2.29}$$

$$FS_{nt}(1:NP\text{-}4)=FS(\text{sorte\_index}(5:NP)). \tag{2.30}$$

Step 7. Determination of the number of available computing resources of the system.

At this stage, the amount of computing resources available for calculations is determined. In accordance with the provisions outlined in Step 4, the concept of updating the provisions of the RA is chosen.

Step 8. Intelligence (surrounding the prey).

The encirclement (reconnaissance) phase is the phase of global exploration of RA space. The RA research strategy is related to the current number of iterations. If $t\leq 0.25T$, the RA will enter the strategy of a high move. The reconnaissance phase is described by the following mathematical expression:

$$Xnew_i^j = \begin{cases} XG^j \times -\eta_i^j \times \beta - R_i^j \times \left(\varsigma \cdot rand\right), t \leq \dfrac{T}{4}; \\ XG^j \times X_{r1}^j \times \left(\varsigma \cdot rand\right) \times ES, t > \dfrac{T}{4} \text{ and } t \leq \dfrac{T}{2}, \end{cases} \tag{2.31}$$

CHAPTER 2

where $Xnew_i^j$ is the j-th dimension of the i-th new RA solution; XG^j is the j-th dimension of the optimal solution obtained at the moment; t is the current iteration number; T is the maximum number of iterations; η_i^j is the j-th hunting operator of the i-th RA, which is calculated using the expression (2.32); β is a constant and is equal to 0.1; R_i^j is the value of the i-th option of the decision, used to reduce the search area by the j-th size, which is calculated using equation (2.33); r_1 randomly takes values from 1 to n; ς is the degree of noise of the data circulating in the system; ES is a random number in the range from –2 to 2, when the number of iterations decreases and equation (2.34) will be used to calculate:

$$\eta_i^j = XG^j \times P_i^j, \tag{2.32}$$

$$R_i^j = \frac{XG^j - X_{r2}^j}{XG^j - \varepsilon}, \tag{2.33}$$

$$ES = 2 \times r_3 \times \left(1 - \frac{1}{T}\right), \tag{2.34}$$

$$P_i^j = \alpha + \frac{X_i^j - Md_i}{XG^j \times \left(UB^j - LB^j\right) + \varepsilon}, \tag{2.35}$$

In equation (2.32), ε is the minimum; r_2 is a number chosen randomly from the range 1–n. In equation (2.33), r_3 is a random integer from –1 to 1. In equation (2.34), α=0.1; Md_i is the average position of the i-th candidate for the solution, which is calculated using equation (2.35):

$$Md_i = \frac{1}{d}\sum_{j=1}^{d} X_i^j, \tag{2.36}$$

where d is the dimensionality of the problem to be solved.

Step 9. Verification of hitting the global optimum. At this stage, the condition for the algorithm to reach the global optimum is checked according to the specified optimization criterion.

Step 10. Global restart procedure.

The restart procedure can effectively improve the ability of the algorithm to go beyond the current optimum and improve the exploratory ability of the algorithm. If the optimal population of the algorithm remains unchanged after *ke* iterations, the population will most likely collapse into a local optimum. Thus, the candidate solution will be initialized randomly to speed up the departure from the global optimum:

$$Xnew_i^j = rand \times \left(UB - LB\right) + LB,\ nt > ke. \tag{2.37}$$

Step 11. Hunting phase (exploitation).

Under the condition t>0.5T, the hunting phase begins, and when t>3/4T and $t \le T$ are the hunting cooperation strategies of the RA. Its equation for updating position is as follows:

$$Xnew_i^j = \begin{cases} XG^j \times P_i^j \times (\varsigma \cdot rand),\ t > \dfrac{T}{2} \text{ and } t \le 3\dfrac{T}{4}; \\ XG^j \times \eta_i^j \times \varepsilon - R_{r1}^j \times (\varsigma \cdot rand),\ t > 3\dfrac{T}{4} \text{ and } t \le T. \end{cases} \quad (2.38)$$

Finally, if the position of the new candidate RA is closer to the food than the current one, the RA will move to the new candidate position and proceed to the next iteration:

$$X_i^j(t+1) = Xnew_i^j(t), \text{ if } F(X_i(t)) > F(Xnew_i(t)), \quad (2.39)$$

where $F()$ is a function to calculate the matching value; X_i is the location of the i-th candidate solution; $Xnew_i$ is the location of the i-th new candidate solution.

Step 12. Combining individual optimization algorithms into a mixed one.

To combine different types of natural optimization algorithms, an ensemble mutation strategy is used, which can generate various individuals to improve the global search capabilities of the hybrid algorithm, which is written as follows:

$$V_{i1} = \begin{cases} X_{R1} + F_1 \times (X_{R2} - X_{R3}),\ r_{10} < C_1; \\ X_i,\ r_{10} \ge C_1, \end{cases} \quad (2.40)$$

$$V_{i2} = \begin{cases} X_{R4} + F_2 \times (X_{R5} - X_{R6}) + F_2 \times (X_{R7} - X_{R8}),\ r_{11} < C_2; \\ X_i,\ r_{11} \ge C_2, \end{cases} \quad (2.41)$$

$$V_{i3} = \begin{cases} X_i + F_3 \times (X_{R9} - X_i) + F_3 \times (X_{R10} - X_{R11}),\ r_{12} < C_3; \\ X_i,\ r_{12} \ge C_3, \end{cases} \quad (2.42)$$

where V_{i1}, V_{i2} and V_{i3} are newly generated mutant positions of the i-th search agent; $R_1 \sim R_{11}$ are the different integer indicators in the range [1, N]; F_1, F_2 and F_3 are scale factors with values of 1.0, 0.8, and 1.0, respectively; $r_{10} \sim r_{12}$ are the random numbers in the range [0, 1]. In addition, the C_1, C_2 and C_3 parameters are 0.1, 0.2, and 0.9, indicating the crossover rate.

After generating candidate mutant positions V_{i1}, V_{i2} and V_{i3}, the best position V_i with the lowest fitness value will be selected to compare with the fitness of the original position X_i, and then the best position will be saved as a new X_i to participate in the next iteration calculation. These processes can be described using equation (2.43):

$$X_i = \begin{cases} V_i, & \text{if } F(V_i) < F(X_i); \\ X_i, & \text{otherwise}, \end{cases} \quad (2.43)$$

where $F(\cdot)$ is the cost function.

Step 13. Checking the stop criterion. The algorithm terminates if the maximum number of iterations is completed. Otherwise, the behavior of generating new places and checking conditions is repeated.

CHAPTER 2

Step 14. Learning RA knowledge bases.

In this research, the learning method based on evolving artificial neural networks developed in the research [2] is used to learn the knowledge bases of each RA.

The method is used to change the nature of movement of each RA, for more accurate analysis results in the future.

Learning the RA knowledge base is an important step that allows agents to accumulate experience and improve their performance in the process of performing tasks. This learning may include storing information about discovered resources, objective functions, optimal routes, the behavior of other agents, and mechanisms that help agents make decisions based on the received data.

The end of the algorithm.

The software implementation of the method of increasing the efficiency of processing various types of data in intelligent management systems of network and server architecture is given below:

```
// reptilian agents.cpp
#include <iostream>
#include <vector>
#include <unordered_map>
#include <thread>
#include <future>
#include <mutex>
#include <cstdlib>
#include <ctime>
class Agent {
public:
   int id;
   double speed;
   double position;
   double bestPosition;
   double bestValue;
   Agent(int id, double speed)
      : id(id), speed(speed), position(0.0), bestPosition(0.0), bestValue(-1.0) {}
   void move() {
      position += speed; }
   void updateBest(double value) {
      if (value > bestValue) {
         bestValue = value;
         bestPosition = position; } }};
class AgentSwarm {private:
```


CHAPTER 2

```
std::vector<Agent> agents;
std::mutex mtx;
double globalBestValue;
double globalBestPosition; public:
AgentSwarm(int numAgents) : globalBestValue(-1.0), globalBestPosition(0.0) {
    std::srand(static_cast<unsigned int>(std::time(nullptr)));
    for (int i = 0; i < numAgents; ++i) {
        double speed = (std::rand() % 10) + 1;
        agents.emplace_back(i, speed); }}
void moveAgents() {
    std::vector<std::future<void>> futures;
    for (auto& agent : agents) {
        futures.emplace_back(std::async(std::launch::async, [&]() {
            std::lock_guard<std::mutex> lock(mtx);
            agent.move();
            double value = evaluate(agent.position);
            agent.updateBest(value);
            updateGlobalBest(agent);
            std::cout << "Agent " << agent.id << " moved to position: " << agent.position
                << ", Value: " << value << std::endl;}));}
    for (auto& future : futures) {
        future.get(); }}
double evaluate(double position) {
    return -1 * (position – 5) * (position – 5) + 10; }
void updateGlobalBest(Agent& agent) {
    if (agent.bestValue > globalBestValue) {
        globalBestValue = agent.bestValue;
        globalBestPosition = agent.bestPosition; }}
void checkGlobalOptimum() {
    std::cout << "Global Best Position: " << globalBestPosition
        << ", Global Best Value: " << globalBestValue << std::endl; }
void globalRestart() {
    std::cout << "Restarting swarm..." << std::endl;
    for (auto& agent : agents) {
        agent.position = 0.0;
        agent.bestPosition = 0.0;
        agent.bestValue = -1.0; }
    globalBestValue = -1.0;
    globalBestPosition = 0.0;}
```

```
    void huntPhase() {
        std::cout << "Hunting Phase Started..." << std::endl;
        moveAgents();  }
    void learnKnowledgeBase() {
        std::cout << "Learning Knowledge Base..." << std::endl; }};
class FoodSource {
public:
    std::string type;
    double value;
    FoodSource(std::string type, double value) : type(type), value(value) {}};
void classifyFoodSources() {
    std::vector<FoodSource> foodSources = {
        FoodSource("Insect", 10.0),
        FoodSource("Plant", 5.0),
        FoodSource("Meat", 20.0)};
    std::cout << "Food sources classified:\n";
    for (const auto& food : foodSources) {
        std::cout << "Type: " << food.type << ", Value: " << food.value << std::endl; }}
int main() {
    int numAgents = 5;
    std::cout << "Number of agents: " << numAgents << std::endl;
    AgentSwarm swarm(numAgents);
    std::cout << "Available computational resources: " << std::thread::hardware_concurrency()
<< " threads" << std::endl;
    classifyFoodSources();
    for (int i = 0; i < 3; ++i) {
        std::cout << "\nIteration " << i + 1 << std::endl;
        swarm.moveAgents();
        swarm.checkGlobalOptimum();
        swarm.huntPhase();
        swarm.learnKnowledgeBase();
        swarm.globalRestart(); }
    return 0;}
```

CONCLUSIONS

1. The research proposed a complex model of processing various types of data in intelligent decision-making support systems.

The essence of the complex approach in modeling is that two partial models are proposed: a model for processing heterogeneous data in intelligent decision-making support systems and a model for processing homogeneous data in intelligent decision-making support systems.

The analysis of the model of the intelligent DMSS for the processing of various types of data allows to draw the following conclusion. While solving the task of processing various types of data, a model of intelligent DMSS is proposed, in contrast to the traditional ones, even in the process of solving partial tasks incorrectly started by experts prb^h with the help of self-organization, expressed in the coordination of partial tasks, the decision maker strives for an ideal solution to the task of processing various types of data. This increases the efficiency of finding an acceptable result from the processing of various types of data prb^u. At the same time, errors in the processing of various types of data will be detected and corrected before the result is obtained prb^u. At the same time, in classic DMSS, a repeated solution is required to detect an error in the processing of various types of data.

The model of homogeneous data processing is based on the idea that the same processing of homogeneous data in intelligent DMSS can be solved in parallel by different functional elements. The relations of integration of elements arise as internal non-verbal images in the memory of the user, who can compare the dynamics of the simulation of the task of processing various types of data in intelligent DMSS from different points of view, which allows to see what modeling using a single model does not provide. Another assumption is developed in the model: the inclusion of a model of a person making a decision in the model of the intelligent DMSS leads to the emergence of the effect of self-organization. This makes it possible to interrelate the results of the work of individual functional elements of the intelligent DMSS in the process of synthesizing the solution to the task of processing various types of data, and not after, as in known models, which ensures the relevance of the intelligent DMSS to the real process of collective problem solving.

2. In this research, a method of processing various types of data in intelligent management systems of network and server architecture is proposed. The essence of the method consists in the processing of various types of data, which are different in nature, with units of measurement due to the use of the duck flock algorithm.

The novelties of the proposed method are:

– the type of uncertainty of the initial data on the state of the intelligent management system of the network and server architecture is taken into account, thereby reducing the time for making the first decision;

– the presence of a procedure for the implementation of an adaptive strategy for the search for food sources of the DA, which allows determining the priority of information processing;

– taking into account the presence of a predator while choosing food sources, which increases the level of information security during its processing;

– taking into account the available computing resources of the intelligent network and server architecture management system, thereby preventing overloading of the intelligent network and server architecture management system.

3. Also, in this research, a method of increasing the efficiency of processing various types of data in intelligent management systems of network and server architecture is proposed. The essence of the method of increasing the efficiency of processing various types of data is to reduce the time of processing various types of data due to the use of an improved algorithm of the herd of reptiles.

The novelties of the proposed method are:

– taking into account the type of uncertainty and noisy data circulating in intelligent network and server architecture management systems. This is achieved through the use of appropriate correction coefficients;

– using the procedure of ensemble mutation, which allows to increase the efficiency of processing various types of data due to the use of the selection of appropriate metaheuristic algorithms and their joint data use;

– changing the search area by individual agents, which is achieved by adjusting the arrival of various types of data;

– changing the speed of movement of agents, which achieves the priority of processing various types of data;

– conducting training of knowledge bases, which is carried out by training the synaptic weights of the artificial neural network, the type and parameters of the membership function, as well as the architecture of individual elements and the architecture of the artificial neural network as a whole;

– avoiding the problem of local extremum.

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

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.

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

CHAPTER 3

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):

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

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:

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

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

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

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; $\bar{Z}(t)$ – 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):

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

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

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.

CHAPTER 3

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.

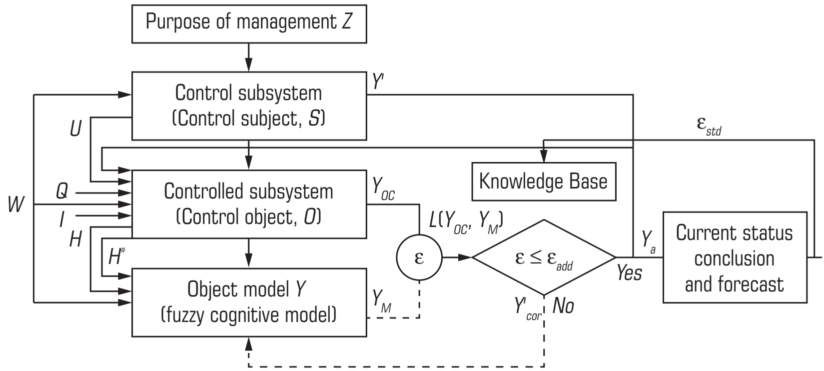

Fig. 3.1 Structural diagram of the system for analyzing and forecasting the state of an object

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 $\varepsilon \leq \varepsilon_{per}$ is not met), the control subsystem adjusts the FCM (Y_{cor}). If $\varepsilon \leq \varepsilon_{per}$ 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:

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

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**).

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:

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

where $x_{v_{i\ \text{nes}}}$ – the current value of the node parameter; $x_{v_{i\ \max}}$, $x_{v_{i\ \min}}$ – 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 $w_{ij}^{*norm}=\frac{w_{ijn}^{norm}}{N}$;

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:

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

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 $S_1^{'} = \left\{ \tilde{s}_1^{(t)}, \tilde{s}_1^{'(t-1)}, \tilde{s}_1^{'(t-2)} \right\}$.

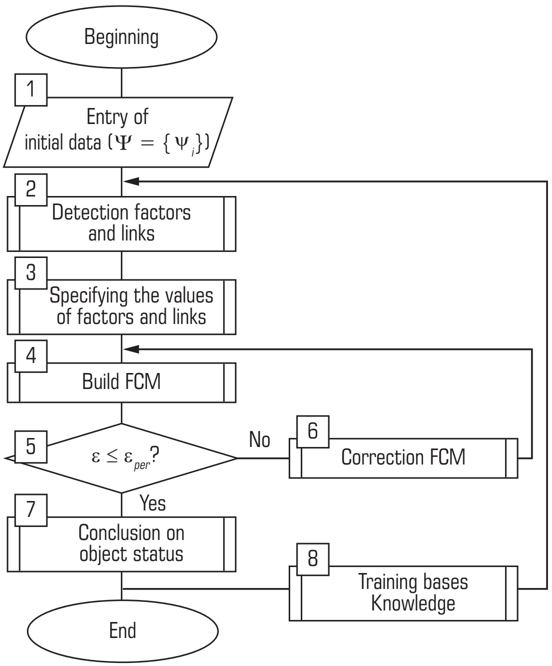

Fig. 3.2 Algorithm for implementing the method of object state analysis and forecasting

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:

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

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):

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

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

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

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:

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

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 $\left\{ w_{ij}^{(t-l_i^j)} \middle| l_i^j = 0,\ldots,L_i^j \right\}$ 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

DOI: 10.15587/978-617-8360-13-9.CH4

Svitlana Kashkevich, Olha Matsyi, Anastasiia Voznytsia,
Oleksiy Buyalo, Daniil Krant, Kostiantin Radchenko

CHAPTER 4

SCIENTIFIC AND METHODOLOGICAL APPARATUS FOR PROCESSING DIVERSE DATA IN AUTOMATED CONTROL SYSTEMS

ABSTRACT

This section of the study proposes the conceptual foundations for the use of artificial intelligence in intelligent decision support systems.

In the course of the research, the authors:

– justified the feasibility of using artificial intelligence theory for processing heterogeneous data in automated control systems;

– developed a methodology for data distribution in automated control systems;

– designed a model for evaluating the process of heterogeneous data processing in automated troop control systems using expert information;

– improved the methodology for configuring an information system to evaluate the process of heterogeneous data processing in automated control systems under conditions of uncertainty.

The analysis conducted in the study established that the application of fuzzy graphs and the mathematical apparatus of fuzzy logic in decision support tasks for data distribution and the evaluation of heterogeneous data processing under various conditions, including uncertainty, enables the distribution of data among elements of automated control systems based on the importance of the elements and the number of features in real-time.

The methodology for rational data distribution based on the importance of automated control system elements and the number of features in such systems under conditions of uncertainty has been improved. This methodology differs from existing ones by combining the mathematical apparatus of information theory, fuzzy logic, and expert evaluation, enabling the formalization of features in a unified parameter space and the intellectualization of information processing processes.

A quantitative assessment of the proposed methodology's efficiency was conducted. The results of this assessment demonstrated that data distribution among the elements of automated control systems based on importance and the number of features using the proposed methodology improves the timeliness of data processing and decision-making regarding the state of the heterogeneous data processing process by 15–17 %.

An enhanced methodology for configuring the information system for evaluating the heterogeneous data processing process in automated control systems under conditions of uncertainty, utilizing

a genetic algorithm, was developed. This methodology addresses limitations of other methods in varying specific features while holding other indicators constant, thus improving the efficiency of the developed information system for evaluating heterogeneous data processing in automated control systems.

The scientific outcome is the improvement of the genetic algorithm for differentiated tuning of the fuzzy knowledge base of the information system for evaluating heterogeneous data processing in automated control systems based on posterior data.

A quantitative assessment of the improved methodology's effectiveness was performed. The results indicated that the proposed methodology enhances the timeliness of configuring the information system for processing heterogeneous data in automated control systems under conditions of uncertainty.

KEYWORDS

Artificial intelligence, heterogeneous data processing, automated control systems, reliability, and timeliness.

4.1 JUSTIFICATION FOR THE FEASIBILITY OF USING ARTIFICIAL INTELLIGENCE THEORY FOR PROCESSING HETEROGENEOUS DATA IN AUTOMATED CONTROL SYSTEMS

The purpose of this section is to justify the necessity of applying the theory of fuzzy graphs to describe the process of heterogeneous data processing in automated control systems (ACS) [1–20].

One of the possible ways to model the process of heterogeneous data processing in ACS is through the application of fuzzy graph theory, whose primary advantage lies in its ability to adequately represent output data in relation to input information that is characterized by weakly structured (fuzzy) indicators. This advantage makes fuzzy graph theory applicable in tasks involving the analysis of operational situations under conditions of uncertainty [21–39].

The model of the heterogeneous data processing process in ACS can be represented in the form of a knowledge matrix (knowledge base) (**Table 4.1**), which contains quantitative and qualitative features and characteristics of ACS functioning [40–59].

A knowledge matrix [60, 61] is a table formed according to the following rules:

1. The dimensionality of the matrix $(n+1)\times N$, where $(n+1)$ – the number of columns, and $N=k_1+k_2+\ldots+k_m$ – the number of rows.

2. The first n the columns of the matrix correspond to the input variables $i=\overline{1,n}$, and $(n+1)$-th a column corresponds to the values dj of the output variable $y\left(j=\overline{1,m}\right)$.

3. Each row of the matrix represents a specific combination of knowledge about the input variables, which has been assigned by an expert to one of the possible values of the output variable y. At the same time: first k_j rows correspond to the value of the output variable $y=d_1$, second k_2 rows correspond to the value $y=d_2$, last k_m – value $y=d_m$.

4. Element a_i^{jp}, that lies at the intersection *i*-th of the column and *jp*-th the row corresponds to the linguistic evaluation of the parameter x_1 in the row of the fuzzy knowledge base with the number *jp*. At the same time, the linguistic evaluation a_i^{jp} is selected from the term set of the corresponding variable x_i, so $a_i^{jp} \in A_i$, $i=\overline{1,n}$, $j=\overline{1,m}$, $p=\overline{1,k_j}$.

Table 4.1 Model of ACS functioning over a time interval

Feature vector number at the input	Features of ACS elements (input variables)				Situation assessment decision (Output variable)
	x_1	x_2	… x_i …	x_n	y
11	a_1^{11}	a_2^{11}	… a_i^{11} …	a_n^{11}	d_1
12	a_1^{12}	a_2^{12}	… a_i^{12} …	a_n^{12}	
…	…	…	…	…	
$1k_1$	$a_1^{1k_1}$	$a_2^{1k_1}$	… $a_i^{1k_1}$ …	$a_n^{1k_1}$	
…					
j_1	a_1^{j1}	a_2^{j1}	… a_i^{j1} …	a_n^{j1}	d_j
j_2	a_1^{j2}	a_2^{j2}	… a_i^{j2} …	a_n^{j2}	
…	…	…	…	…	
jk_j	$a_1^{jk_j}$	$a_2^{jk_j}$	… $a_i^{jk_j}$ …	$a_n^{jk_j}$	
…					
$m1$	a_1^{m1}	a_2^{m1}	… a_i^{m1} …	a_n^{m1}	d_m
$m2$	a_1^{m2}	a_2^{m2}	… a_i^{m2} …	a_n^{m2}	
…	…	…	…	…	
mk_m	$a_1^{mk_m}$	$a_2^{mk_m}$	… $a_i^{mk_m}$ …	$a_n^{mk_m}$	

The presented model structurally consists of layers of features (sets of input informational arrays) at specific time intervals and possible variants of ACS functioning (sets of decisions). Decision-making is performed at each stage, taking into account the features of ACS functioning [62–86].

Hierarchical Fuzzy Inference Systems. For modeling multidimensional "input-output" dependencies, it is advisable to use hierarchical fuzzy inference systems. In such systems, the output of one knowledge base serves as the input to another, higher-level hierarchy. Hierarchical knowledge bases lack feedback loops. **Fig. 4.1** presents an example of a hierarchical fuzzy system that models the dependency $y=f(x_1, x_2, x_3, x_4, x_5, x_6)$ using three knowledge bases f_1, f_2, f_3.

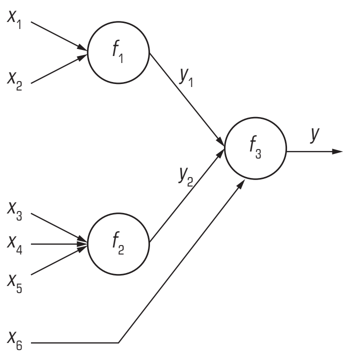

Fig. 4.1 Example of a hierarchical fuzzy inference system

These knowledge bases describe dependencies $y_1=f_1(x_1, x_2)$, $y_2=f_2(x_4, x_5, x_6)$ and $y=f_3(y_1, x_3, y_2)$. The application of hierarchical fuzzy knowledge bases allows overcoming the "curse of dimensionality". Another advantage of hierarchical knowledge bases is their compactness. A small number of fuzzy rules in hierarchical knowledge bases can adequately describe multidimensional "input-output" dependencies.

Let's assume that five terms are used for the linguistic evaluation of variables. In this case, the maximum number of rules required to define the dependency $y=f(x_1, x_2, x_3, x_4, x_5, x_6)$ using a single knowledge base will amount to $5^6=15625$. In fuzzy inference using a hierarchical knowledge base, the defuzzification and fuzzification procedures for intermediate variables y_1 and y_2 (**Fig. 1**) are not performed. The result of the logical inference in the form of a fuzzy set $\tilde{y}^*=\left(\frac{\mu_1(X^*)}{\tilde{d}_1}, \frac{\mu_2(X^*)}{\tilde{d}_2}, \frac{\mu_m(X^*)}{\tilde{d}_m}\right)$ is directly transmitted to the fuzzy inference engine of the next level in the hierarchy. Therefore, for intermediate variables in hierarchical fuzzy knowledge bases, it is sufficient to define only the term sets without describing the membership functions.

4.2 METHODOLOGY FOR DATA DISTRIBUTION IN AUTOMATED CONTROL SYSTEMS

The essence of the data distribution methodology in automated control systems (ACS) lies in the rational allocation of data among ACS elements.

By selecting the most critical data sources and an optimal number of gradations, it ensures the desired probability of correctly identifying the type of data circulating within the system.

To choose the best plan for distributing data sources among ACS elements, the following partial quality indicators for distribution are used:

1. Completeness of ACS Element Coverage by Observation: this is calculated as the ratio of the sum of importance coefficients of ACS elements Y_j included in the distribution plan to the sum of importance coefficients of all ACS elements:

$$\Pi = \frac{\sum_{j=1}^{\{m\}_u} Y_j}{\sum_{j=1}^{J} Y_j}, \tag{4.1}$$

where $\{m\}_u$ – the number of ACS elements selected for observation in u-th distribution plan.

2. The expenditure of technical resources for ACS load, determined as the sum of the technical resource expenditures of the ACS $S_{sum_{ACS}}$.

3. The probability of tracking the status and nature of activities of the entire set of ACS elements subject to distribution – $\overline{P}$:

$$\overline{P} = \frac{1}{m}\sum_{j=1}^{m} P_j, \tag{4.2}$$

where P_j – the probability of tracking the status and nature of activities of ACS elements included in the distribution plan.

Then, the system of partial quality indicators for selecting a data distribution plan among ACS elements will have the following form:

$$\begin{cases} \Pi \to \max; \\ S_{sum_{ACS}} \to \min; \\ \overline{P} \to \max. \end{cases} \tag{4.3}$$

Taking into account the system of partial indicators, the functional reflecting the quality of data distribution among ACS elements, depending on the selection of a specific distribution plan option Π, can be expressed as:

$$F_{\Pi_{U_{opt}}} = \max F(\Pi_{\Pi_U}, S_{sum_{ACS_{\Pi_U}}}, \overline{P}_{\Pi_U}). \tag{4.4}$$

However, in existing methodologies for information distribution based on the importance of ACS elements, the calculation of importance coefficients is performed implicitly (4.4), and the procedure for their calculation is not defined.

Thus, a relevant scientific problem arises: multi-criteria optimization of the data distribution process among ACS elements, considering their importance, to improve data processing efficiency.

Then, (4.4) can be rewritten as:

$$F_{\Pi_{U_{opt}}} = \arg\max_{im \in Im} F(Im, S_{sum_{ACS_{\Pi_U}}}, \bar{P}_U), \tag{4.5}$$

where Im – the vector of importance (priority) coefficients of ACS elements in the observation range; P_{Π_U} – the probability of tracking the status and nature of an ACS element's activity when selecting u-th a distribution plan.

The importance of an ACS element can be considered as a non-metric utility criterion (NMUC). The main challenge in solving this problem lies in representing the NMUC in a quantitative form for its subsequent integration into a utility function (UF).

To represent the NMUC in a quantitative form, non-metric partial utility criteria (NMPC) have been defined to characterize the importance of an ACS element.

The main NMPCs include the degree of task priority for which the distribution is carried out or the priority level of the ACS element (X_{pr}); the degree of informational value (X_{inf}); the degree of operational value of the ACS element (X_{op}).

Let $Q(X_{pr}, X_{inf}, X_{op})$, denote the utility function of the NMPC. X_{pr}, X_{inf}, X_{op} independent systems of values. Then the utility functions of the NMPC can be represented by the following system of expressions:

$$\begin{cases} \psi_{pr} = Q(X_{pr})f(X_{pr}), \\ \psi_{inf} = Q(X_{inf})f(X_{inf}), \\ \psi_{op} = Q(X_{op})f(X_{op}), \end{cases} \tag{4.6}$$

where $f(X_{pr}), f(X_{inf}), f(X_{op})$ – functions of utility dependence on metric criteria.

In turn, the utility function of an ACS element will be expressed as:

$$\psi = Q(X_{pr}, X_{inf}, X_{op})f(X_{pr}, X_{inf}, X_{op}). \tag{4.7}$$

To study the impact of non-metric criteria, let's introduce a constraint, the essence of which is that the influence of metric criteria is equivalent, meaning there is no dependence on metric criteria:

$$f(X_{pr}) = f(X_{inf}) = f(X_{op}) = 1. \tag{4.8}$$

Analysis of the constraint (4.8) reveals that the indicators are equivalent to each other concerning the metric criterion. In turn, the utility dependence functions on non-metric criteria vary linearly and are determined by the lower and upper values of the accepted evaluations. By performing normalization based on the maximum value and the adopted scale, any preference for one of the indicators in expression (4.8) concerning a non-metric criterion will lead to the dominance of the utility function of the corresponding indicator.

Considering (4.8), expression (4.7) can be represented as:

$$\psi = Q(X_{pr}, X_{inf}, X_{op}). \tag{4.9}$$

For the purpose of selecting the rational form of the utility function $Q(X_{pr}, X_{inf}, X_{op})$ it is convenient to represent X_{pr}, X_{inf}, X_{op} in the form of fuzzy sets [63, 64], with NMPC evaluations as their elements, respectively. Then $Q(X_{pr}, X_{inf}, X_{op})$ it can be identified with the membership function of the set of input values of the primary NMPC indicators x_{pr}, x_{inf}, x_{op} to fuzzy sets X_{pr}, X_{inf}, X_{op}, accordingly. Thus, the task of determining the importance of an ACS element can be formulated as a decision-making problem regarding the importance of the ACS element, and the result of the decision-making process can be represented as:

$$\mathrm{Im} = Q(x_{pr}, x_{inf}, x_{op}), \tag{4.10}$$

where x_{pr}, x_{inf}, x_{op} – a set of input values of the primary NMPC indicators; Im – a decision regarding the determination of the importance of the ACS element.

The task of decision-making regarding the determination of the importance of the ACS element is to, based on information about the vector of input indicators $(x_{pr}, x_{inf}, x_{op})$ determine the outcome Im. A necessary condition for the formal solution of the stated problem is the presence of the dependency (4.10). To establish such a dependency, it is necessary to consider the input indicators (NMPC) and the output decision as linguistic variables defined on universal sets.

To evaluate such linguistic variables, it is proposed to use qualitative terms that form a term set:

$X_{inf} = \{L, bA, M, aA, H\}$ – the term set of a variable x_{inf},

$X_{op} = \{L, bA, M, aA, H\}$ – the term set of a variable x_{op},

$X_{pr} = \{L, M, H\}$ – the term set of a variable x_{pr},

$\mathrm{Im} = \{L, bA, M, aA, H\}$ – the term set of a variable Im,

where L, bA, M, aA, H – respectively "low", "below average", "average", "above average", "high"; Im – the set of variables characterizing the importance of an ACS element:

$$\begin{aligned} X_{inf} &= [1,5], \\ X_{op} &= [1,5], \\ X_{pr} &= [1,3], \\ \mathrm{Im} &= [1,5]. \end{aligned} \tag{4.11}$$

To evaluate the values of linguistic variables x_{pr}, x_{inf}, x_{op}, in accordance with (4.11), let's use the corresponding scale of qualitative terms.

In accordance with cognitive engineering methods for knowledge base synthesis, knowledge bases have been developed that characterize the importance of elements. Using the mathematical apparatus of fuzzy set theory, the knowledge base is transformed into logical equations.

$$\mu^{\mathrm{Im}_j}(X_{pr}, X_{inf}, X_{op}) = \max_J \left\{ \min_i \left[\mu^J(x_{i(pr)}), \mu^J(x_{i(inf)}), \mu^J(x_{i(op)}) \right] \right\}, \tag{4.12}$$

where μ – the membership functions of the corresponding linguistic variables x_{pr}, x_{inf}, x_{op}, Im, the sets X_{pr}, X_{inf}, X_{op}, Im, $J \in \{L, bA, M, aA, H\}$; $x_{i(pr)} \in [1,2,3]$; $x_{i(inf)} \in [1,2,3,4,5]$; $x_{i(op)} \in [1,2,3,4,5]$; $\mathrm{Im}_i \in [1,2,3,4,5]$.

From the analysis of the numerical results of the experiment, it was concluded that the decision regarding the importance of the elements of the automated control system (ACS) is determined by the expression:

$$\frac{x_{inf} + x_{op}}{2} \cdot x_{pr} = \mathrm{Im}. \tag{4.13}$$

The minimum value that expression (4.13) can take is Im=1, then the maximum value Im=15. Let's define the FN (membership function) for the term sets of the importance of the elements of the ACS.

For this, it is possible to normalize the measurement intervals of each variable to a single universal interval [0, 4] using the following relationship:

$$\mu^j(\mathrm{Im}_i) = \tilde{\mu}^j(u), u = 4\frac{\mathrm{Im}_i - \underline{\mathrm{Im}}}{\overline{\mathrm{Im}} - \underline{\mathrm{Im}}}, j = L, bA, M, aA, H. \tag{4.14}$$

The analytical model of the membership function is represented by the expression:

$$\mu^j(u) = \frac{1}{1 + \left(\frac{u-b}{c}\right)^2}, \tag{4.15}$$

where the parameters b and c are set based on the results of the previous assessment of the functioning of the ACS.

The graphical representation of the membership function according to expression (4.15) is shown in **Fig. 4.2.**

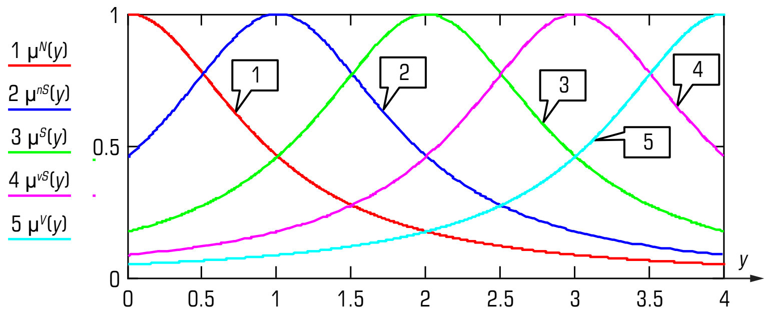

Fig. 4.2 Graphical representation of the membership function of the fuzzy set of the importance degrees of an element in the automated control system (ACS)

CHAPTER 4

Calculation of coefficients PI. Priority coefficient is determined x_{pr}, coefficient of the degree of informational value x_{inf}, coefficient of the degree of operational value x_{op}.

Coefficient of the degree of operational value OP. Using (4.13), the values of the importance of the corresponding element of the ACS are calculated.

Determination of the linguistic value of the importance of the ACS element.

Optimization of the feature vector of the functioning of the ACS element.

It is known that as the number of features that characterize the functioning of the ACS element increases, the time required for identifying its operational mode and other costs, primarily hardware costs, also increase, which in turn reduces the operational efficiency of the evaluation process.

Identifying informative features in a real situation is a complex task, especially when evaluating the functioning process of system elements in an ACS, where the feature set can be very large and the features themselves may be correlated with each other. Therefore, the task is to select and extract the most informative features to reduce the dimensionality of the input data vector, while simultaneously finding a coordinate system in which the probability of correctly recognizing ACS elements will be maximized or sufficient for decision-making.

Reducing the dimensionality of the feature space in the presence of a large number of ACS system elements plays a significant role, as it increases the throughput of the ACS system's channels as a whole. This is because an increase in the number of features that characterize an ACS element significantly leads to an increase in identification errors.

The dependence of the identification probability of the functional process of an ACS element on the dimensionality of the feature space is shown in **Fig. 4.3**.

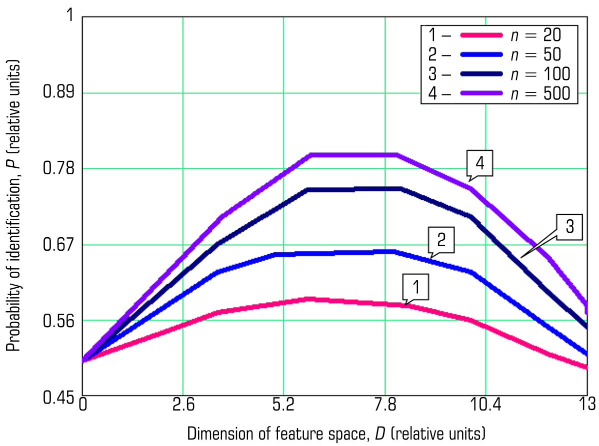

Fig. 4.3 The dependence of the identification probability of the operational mode of an ACS element on the dimensionality of the intelligence feature

The graphs show that arbitrary increases in the dimensionality of the feature space may lead to a deterioration in the probability of correct recognition.

The formation of the feature vector for the ACS element can be mathematically represented as:

$$Y = AX, \tag{4.16}$$

where X – the feature vector that characterizes the operation process of the ACS element; Y – the vector of possible decisions; A – the transformation matrix.

Checking the optimal dimensionality of the feature vector. The condition for the termination of the grading elimination cycle is the value of the informativeness loss threshold for all features $\left(\sum \Delta I_k\right)_{\max}$ or for a specific feature. It is also possible to set a maximum number of gradations that need to be retained during the minimization process.

Fig. 4.2 shows the curves of changes in the informativeness of features depending on the number of their gradations. Analysis of these curves for all features allows minimizing the number of gradations in terms of the memory usage of the identification device and the total loss of informativeness for identification parameters. The presented dependencies suggest that the number of gradations for features should be chosen to be no more than 4–6 (at the inflection point of most curves), which coincides with the results presented in **Fig. 4.3**.

The adjustment of system parameters is carried out based on an improved methodology, which is based on the use of a genetic approach. This method facilitates the correction of the system parameters.

4.3 MODEL OF THE PROCESS OF EVALUATING THE PROCESSING OF HETEROGENEOUS DATA IN AN AUTOMATED CONTROL SYSTEM (ACS) USING EXPERT INFORMATION

Let the following be known: the set of solutions $D = \{d_j\}$, $\left(j = \overline{1,m}\right)$, that corresponds to the result of evaluating the processing of various types of data in an automated control system (ACS) y; the set of input indicators $X = \{x_i\}$, $\left(i = \overline{1,n}\right)$; the ranges of quantitative variation for each input information; membership functions that allow representing the indicators $x_i \in \left[\underline{x_i}, \overline{x_i}\right]$, $i = \overline{1,n}$, x_i, $i = \overline{1,n}$ in the form of fuzzy sets; a knowledge matrix defined by rules (**Table 4.1**). It can be graphically represented as shown in **Fig. 4.4**.

Let's consider the application of the model for utilizing expert information to synthesize an algorithm for evaluating the processing of various types of data in an automated control system (ACS).

From the analysis of the functioning of ACS elements under various situational conditions, the evaluation directions have been identified: the similarity of situational indicators and their changes during the operation of the ACS.

Let's describe the model for evaluating the processing of various types of data in an automated control system (ACS):

$$D(k) = f\begin{bmatrix} Y_1(k-1), Y_2(k-1), \ldots, Y_{14}(k-1), Q(k-1), R(k-1), \\ Z_1(k-1), \ldots, Z_4(k-1) \end{bmatrix}, \tag{4.17}$$

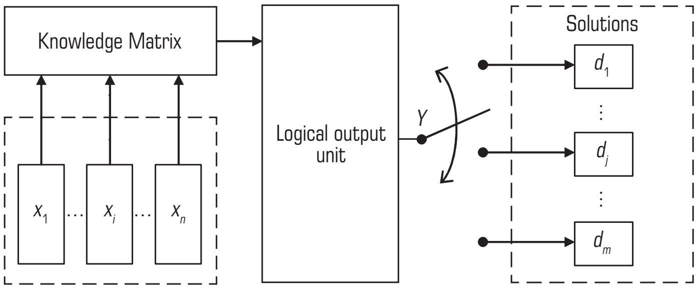

Fig. 4.4 Model of the operational situation assessment process

where $Y_1(k-1)$ – a vector that characterizes the operating mode of ACS element No. 1 at k–1 modeling step; $Y_2(k-1)$ – a vector that characterizes the operating mode of an ACS element at k–1 modeling step; $Y_{14}(k-1)$ – a vector that characterizes the operating mode of ACS element No. 14 at k–1 modeling step; $Q(k-1)$ – a vector that characterizes the operating mode of the control and communication system of ACS element No. 1; $R(k-1)$ – a vector that characterizes the operating mode of the control and communication system of an ACS element; $Z_1(k-1),\ldots,Z_4(k-1)$ – vectors that characterize the operating modes of control and communication systems of group ACS elements.

In turn, the vectors of the data processing evaluation process in an ACS are determined by the following indicators: $Y_1,\ldots,Y_{14},Q,R,Z_1,\ldots,Z_4=\{k_{11}(x),\ldots,k_{145}(x)\}$.

For indicators with quantitative measurements, the range of variation is divided into four quanta. This ensures the possibility of transforming a continuous universal set $U=[\underline{u},\overline{u}]$ into a discrete five-element set:

$$U=\{u_1,u_2,\ldots,u_5\},$$

where $u_1=\underline{u}$, $u_2=\underline{u}+\Delta_1$, $u_3=u_2+\Delta_2$, $u_4=u_3+\Delta_3$, $u_5=\overline{u}$, and $\Delta_1+\Delta_2+\Delta_3+\Delta_4=\overline{u}-\underline{u}$, $\overline{u}(\underline{u})$ – the upper (lower) boundary of the indicator's range of variation. Thus, all pairwise comparison matrices have a dimension. The choice of four quanta is determined by the possibility of approximating nonlinear curves through five points.

For evaluating the values of linguistic variables, it is possible to use the following scale of qualitative terms.

In the general case, the input variables $x_1, x_2, \ldots, x_n$ can be defined as a number, a linguistic term, or based on the thermometer principle [51, 56].

The evaluation of the data processing process in an ACS using expert information is carried out using fuzzy logical equations, which represent a knowledge matrix and a system of logical statements. These equations allow for the calculation of membership function values for various identification results at fixed input indicator values. As the outcome of the evaluation process for data processing in an ACS, the decision with the highest membership function value will be accepted.

Linguistic evaluations $\underline{\alpha_i^{jp}}$ variables $x_1, x_2, \ldots, x_n$, that are part of the logical statements regarding decisions $d_j, j=\overline{1,m}$, will be considered as fuzzy sets defined on universal sets $X_i = \left[\underline{x_i}, \overline{x_i}\right], i=\overline{1,n}.$

So $\mu^{a_i^{jp}}(x_i)$ – (MF) of the indicator $x_i \in \left[\underline{x}, \overline{x}\right]$ to a fuzzy term $\alpha_i^{jp}, i=\overline{1,n}, j=\overline{1,m}, p=\overline{1,l_j}$; $\mu^{d_j}(x_1, x_2, \ldots, x_n)$ – (MF) of the input variables vector $X=(x_1, x_2, \ldots, x_n)$ to the value of the output evaluation $y=d_j, j=\overline{1,m}$.

The relationship between these functions is determined by a fuzzy knowledge base and can be represented in the form of the following logical equations:

$$\begin{gathered}
\mu^{d_j}(x_1, x_2, \ldots, x_n) = \mu^{a_1^{j1}}(x_1) \wedge \mu^{a_2^{j1}}(x_2) \wedge \ldots \wedge \mu^{a_n^{j1}} \vee \\
\mu^{a_1^{j2}}(x_1) \wedge \mu^{a_2^{j2}}(x_2) \wedge \ldots \wedge \mu^{a_n^{j2}}(x_n) \ldots \\
\ldots \mu^{a_1^{jl_j}}(x_1) \wedge \mu^{a_2^{jl_j}}(x_2) \wedge \ldots \wedge \mu^{a_n^{jl_j}}(x_n), j=\overline{1,m}.
\end{gathered} \tag{4.18}$$

The equations are derived from the fuzzy knowledge base by replacing variables (linguistic terms) with their membership functions (MFs), and the AND and OR operations with the respective operations $\wedge$ and $\vee$.

Briefly, the system (4.18) can be written as follows:

$$\mu^{d_j}(x_i) = \bigvee_{p=1}^{l_j}\left[\bigwedge_{i=1}^{n} \mu^{a_i^{jp}}(x_i)\right], j=\overline{1,m}. \tag{4.19}$$

Fuzzy logical equations are an analog of the fuzzy inference procedure introduced by Zadeh, which is performed using the "fuzzy (min-max) composition" operation, where the $\wedge$ and $\vee$ operations correspond to the min and max operations. From (4.19), let's obtain:

$$\mu^{d_j}(x_i) = \max_{p=\overline{1,l_j}}\left\{\min_{j=\overline{1,n}}\left[\mu^{a_i^{jp}}(x_i)\right]\right\}. \tag{4.20}$$

From expression (4.20), it is evident that for the calculations it is only necessary to have the membership functions (MFs) of the variables to the fuzzy terms.

4.4 ENHANCED METHODOLOGY FOR CONFIGURING THE INFORMATION SYSTEM FOR EVALUATING THE PROCESS OF PROCESSING VARIOUS TYPES OF DATA IN AN ACS UNDER CONDITIONS OF UNCERTAINTY

The essence of the methodology for configuring the information system for evaluating the process of processing various types of data in an ACS under conditions of uncertainty lies in selecting the weight coefficients of production rules, minimizing the error between the reference and experimental decisions.

Identification based on fuzzy logical inference is carried out in accordance with the defined knowledge base:

IF $(x_1 = a_{1,j1})$ AND $(x_2 = a_{2,j1})$ AND ...AND $(x_n = a_{n,j1})$ with weight w_{j1},

OR $(x_1 = a_{1,j2})$ AND $(x_2 = a_{2,j2})$ AND ...AND $(x_n = a_{n,j2})$ with weight w_{j2},

OR $(x_1 = a_{1,jk_j})$ AND $(x_2 = a_{2,jk_j})$ AND ...AND $(x_n = a_{n,jk_j})$ with weight w_{jk_j}, (4.21)

THEN $y = d_j, j = \overline{1,m}$,

where $a_{i,jp}$ – the fuzzy term that evaluates the variable x_i in the row with number $jp(p = \overline{1,k_j})$, i.e., $a_{i,jp} = \int \mu_{jp}(x_i)/x_i$; k_j – the number of rows-conjunctions in which the output y is evaluated by the value d_j; $w_{jp} \in [0,1]$ – the weight coefficient of the rule with the number jp.

Functions of correspondence in the process of handling different types of data $X^* = (x_1^*, x_2^*, \ldots, x_n^*)$ are calculated for the classes d_j as follows:

$$\mu_{d_j}(X^*) = \underset{p=\overline{1,k_j}}{\vee} w_{jp} \cdot \underset{i=\overline{1,n}}{\wedge}\left(\mu_{jp}(x_i^*)\right), j = \overline{1,m}, \tag{4.22}$$

where $\mu_{jp}(x_i^*)$ – the input correspondence function x_i^* an unclear term $a_{i,jp}$; $\wedge(\vee)$ – s-norm (t-norm), which in classification tasks usually corresponds to the maximum (minimum).

As a solution, the class with the maximum matching function of the calculated solution is selected $d_1 \ldots d_m$:

$$y^* = \underset{\{d_1, d_2, \ldots, d_m\}}{\arg} \max\left(\mu_{d_1}(X^*), \mu_{d_2}(X^*), \ldots, \mu_{d_m}(X^*)\right). \tag{4.23}$$

Thus, the adaptation or adjustment of the information system for evaluating the processing of various types of data under uncertainty conditions will be performed.

The work applies an adaptation method based on solving the optimization problem using the genetic algorithm method.

Let's introduce the constraint that there is a reference sample from M a pair of experimental data that link the inputs $X = (x_1, x_2, \ldots, x_n)$ with the output y of the dependency being studied:

$$(X_r, y_r), (r = \overline{1,M}), \tag{4.24}$$

where $X_r = (x_{r,1}, x_{r,2}, \ldots, x_{r,n})$ – the input vector in r-th pair; y – the corresponding output.

Tuning the model involves finding such parameters of the matching functions for the input variable terms and the weighting coefficients of the rules that minimize the deviation between the expected and obtained results on the reference sample. The proximity criterion can be defined in various ways.

The first method involves selecting the classification error percentage on the reference sample used for training the system as the tuning criterion. Let's introduce the following notation:

P – the vector of parameters of the matching functions for the input and output variables;

W – the vector of the weighting coefficients of the knowledge base rules;

$F\left(X_r,P,W\right)$ – the output result according to the knowledge base with the parameters $\left(P,W\right)$ with the input values X_r.

Then the tuning of the fuzzy model is reduced to the following optimization problem: find such a vector $\left(P,W\right)$, to:

$$\frac{1}{M}\sum_{r=\overline{1,M}}\Delta_r \to \min, \tag{4.25}$$

where Δ_r – classification error of the state of processing various types of data X_r:

$$\Delta_r = \begin{cases} 1, if \quad y_r \neq F\left(X_r,P,W\right); \\ 0, if \quad y_r = F\left(X_r,P,W\right). \end{cases} \tag{4.26}$$

The advantage of the tuning criterion lies in its simplicity and clear substantive interpretation. The error percentage is widely used as a training criterion for various pattern recognition systems.

The objective function of the optimization problem takes on discrete values, which complicates the use of gradient optimization methods. It is particularly difficult to select the necessary parameters of gradient algorithms (for example, the increment of arguments for calculating partial derivatives) when tuning a fuzzy classifier on a small data sample.

The second method involves using the distance between the output in the form of a fuzzy set as the tuning criterion $\left(\frac{\mu_{d_1}\left(x\right)}{d_1},\frac{\mu_{d_2}\left(x\right)}{d_2},\dots,\frac{\mu_{d_m}\left(x\right)}{d_m}\right)$ and the value of the output variable in the reference sample, which is intended for system training. For this purpose, the output variable of the reference sample (4.23) is fuzzified as follows:

$$\left.\begin{array}{l} \tilde{y} = \left(1/d_1, 0/d_2,\dots,0/d_m\right), if \quad y = d_1 \\ \tilde{y} = \left(0/d_1, 1/d_2,\dots,0/d_m\right), if \quad y = d_2 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ \tilde{y} = \left(0/d_1, 0/d_2,\dots,1/d_m\right), if \quad y = d_m \end{array}\right\}. \tag{4.27}$$

In this case, tuning the fuzzy classifier is reduced to the following optimization problem: find such a vector (P,W), so that:

$$\frac{1}{M}\cdot\sum_{r=1}^{M}\sum_{j=1}^{m}\left(\mu_{d_j}(y_r)-\mu_{d_j}(X_r,P,W)\right)^2 \to \min, \quad (4.28)$$

where $\mu_{d_j}(y_r)$ – membership function of the output variable value y in r-th the pair of the reference sample to the decision d_j; $\mu_{d_j}(X_r,P,W)$ – membership function of the fuzzy model output with parameters (P,W) to the decision d_j, with the input values from r-the pair of the reference sample (X_r).

The objective function in problem (4.27) does not have extensive plateaus, so it is suitable for gradient-based optimization methods. However, the optimization results are not always satisfactory: the fuzzy knowledge base that ensures the minimum of criterion (4.27) does not always also ensure the minimum classification errors. This is explained by the fact that points close to the maxima of the class partitions usually make the same contribution to the tuning criterion, both in the case of correct classification and in the case of misclassification.

The third method inherits the advantages of the previous methods. The idea is to increase the contribution of misclassified objects to the tuning criterion by multiplying the distance $\sum_{j=1}^{m}\left(\mu_{d_j}(y_r)-\mu_{d_j}(X_r,P,W)\right)^2$ by a penalty coefficient. As a result, the optimization problem takes the form:

$$\frac{1}{M}\cdot\sum_{r=1}^{M}\left(\Delta_r\cdot penalty+1\right)\cdot\sum_{j=1}^{m}\left(\mu_{d_j}(y_r)-\mu_{d_j}(X_r,P,W)\right)^2 \to \min, \quad (4.29)$$

where $penalty > 0$ – penalty coefficient.

Problems (4.26), (4.28), and (4.29) can be solved by various optimization technologies, among which the method of steepest descent, quasi-Newton methods, and genetic algorithms are often used.

Usually, constraints are imposed on the controlled variables to ensure the linear ordering of the elements of the term sets. In addition, the cores of the fuzzy sets must not go beyond the ranges of variation of the corresponding variables. This ensures transparency, that is, meaningful interpretability of the fuzzy knowledge base after tuning. As for the vector W, its coordinates must lie in the range [0,1]. If the requirements for the interpretability of the knowledge base are high, the rule weights are not tuned, leaving them equal to 1. There is also an intermediate option where the weighting coefficients can take values of 0 and 1. In this case, a zero value of the weighting coefficient is equivalent to excluding the rule from the fuzzy knowledge base.

Parameters of the matching functions and rule weights can be tuned simultaneously or separately. When only the rule weights are tuned, the computational volume can be significantly reduced, since the membership functions $\mu_{jp}\left(x_i^*\right)$, do not depend on W. For this, at the beginning of the optimization, it is necessary to calculate the degrees of rule execution with unit weighting coefficients $(w_{jp})=1$ for each object in the reference sample:

$$g_{jp}\left(X_r\right) = \underset{i=\overline{1,n}}{\wedge} \mu_{jp}\left(x_{r,i}\right), j = \overline{1,m}, p = \overline{1,k_j}, r = \overline{1,M}.$$

For the new weighting coefficients of the membership functions for the process of processing various types of data in the ACS X_r classes d_j are calculated as follows:

$$\mu_{d_j}\left(X_r\right) = \underset{p=\overline{1,k_j}}{\vee} w_{jp} \cdot g_{jp}\left(X_r\right), j = \overline{1,m}.$$

Considering the specifics of the process of processing various types of data in an ACS, one of the ways to solve it based on fuzzy logic is to apply combined optimization methods that combine the advantages of the gradient method and the random search method. One such method is the genetic algorithm, which makes it possible to perform optimization for multimodal, non-smooth, and non-convex functions with a convergence speed greater than that of random search methods.

Thus, taking into account the identified features of the process of processing various types of data in an ACS, the hierarchical nature of the logical inference tree, and the structural-semantic model of processing various types of data in an ACS, it is advisable to carry out the process of tuning the information system for processing various types of data in a differentiated manner, i.e., by tuning the fuzzy knowledge base of each individual element of the ACS system.

Let's explore the possibilities of applying and functioning of the "genetic algorithm" for tuning the information system for assessing the operational situation.

Let's assume the following initial data are known:

S – the system structure vector that defines the system parameters that do not change during optimization (regarding the information system for assessing the operational situation – a set of IF-THEN rules represented using the mathematical apparatus of fuzzy logic, and the rule weight coefficients);

B – the reference vector that contains a set of sample stimulus-response pairs (input indicators – decisions) by which the information system for assessing the operational situation is tuned;

W_{jp} – the vector of the rule weights of the fuzzy knowledge base, whose value is being optimized;

F – the mismatch function that determines the quality of the solution proposed by the information system for assessing the operational situation compared to the solution in the reference vector;

F_{ACS} – the mismatch function that determines the quality of the solution proposed by the information system for processing various types of data regarding a separate element of the ACS, compared to the solution in the reference vector.

By decision-making in the information system for processing various types of data in an ACS, let's understand the output result provided by the system according to the entered features $x_{11}...x_{nm}$.

Let's introduce the following constraints:

– the vector B covers the entire practically significant range of solutions within the application domain of the information system for processing various types of data in the ACS;

– the vector S (IF-THEN rules) is formed in advance based on the results of working with experts and does not contain logical errors;

– based on the results of working with experts, the values of the membership function parameters have been determined;

– the mismatch function F calculates the decision-making error of the information system for processing various types of data in an ACS by the least squares method with an ordinal scale according to the formula:

$$e_{ACS} = \sum_{j=1}^{n} \sum_{i=1}^{k} (d_i^{B_j} - d_i^{O_j})^2, \tag{4.30}$$

where e_{ACS} – the tuning error of a separate ACS element; n – the dimension of the vector B; k – the maximum number of solutions issued by the information system for processing various types of data in the ACS; $d_i^{B_j}$ – the reference i-th decision for the j-th input element of vector B; $d_i^{O_j}$ – i-th the solution of the information system for processing various types of data in the ACS for j-th the input element of vector B, taking into account the rule weights W_{jp}; i – the decision number issued by the information system for assessing the operational situation, $i \in \overline{1,k}$; j – the number of the input indicator set in the reference vector B.

For tuning the knowledge base set (KBS) of a separate element of the ACS system, the proposed criterion is:

$$e_{ACS\_\min} = Min(F_{ACS}(S,W)), \tag{4.31}$$

where $e_{ACS\min}$ – the minimally acceptable final total error, as the difference between the membership function values of the decision regarding the state of operation of a separate element of the ACS system and the reference decision.

As a result of using criterion (4.30) for each separate element of the ACS, it is logical to use criterion (4.31), which indicates the tuning of the information system for processing various types of data in the ACS as a whole:

$$e_{\Sigma\min} = Min(F(S,W)), \tag{4.32}$$

where $e_{\Sigma\min}$ – the minimally acceptable final total error as the difference between the membership function values of the decision regarding the state of operation of the control and communication system element of the ACS and the reference decision.

Tuning the rule weights (W_{jp}) and the parameters of the membership functions – vector P, will be performed using the "genetic algorithm" method.

The set of indicators being optimized is combined into a parameter vector called a chromosome. Indicators in the chromosome can be stored in a regular or encoded (transformed) form.

A specific section of the chromosome responsible for encoding a single indicator is called a gene; the length of a gene depends on the chosen type of encoding.

Each chromosome represents a solution to the problem, which is optimized with an efficiency expressed by a certain number calculated using the objective function. A set of chromosomes (a collection of solutions) is called a population. The population maintains a constant number of chromosomes. The main stages of the genetic algorithm method are shown in **Fig. 4.5**.

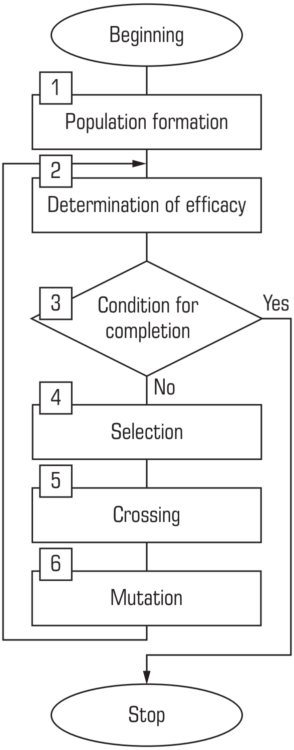

Fig. 4.5 Workflow diagram of the genetic algorithm tuning method

The formation of the initial population depends on the approach to forming chromosomes: tuning the weighting coefficients of the importance of the situation; tuning the weighting coefficients of the priority of logical rules in the knowledge base (KB) based on which decisions are made.

To reduce the volume of data arrays, it is proposed to form chromosomes where the genes are the weighting coefficients of the logical rules in the knowledge base (KB).

Formation of the initial population. The reference vector B is formed based on the results of assessing the operational situation under various conditions of operation of the ACS system elements.

Let's define the membership functions (MF) of the operational features of the ACS system element and decisions to fuzzy terms using the formula:

$$\mu^{d_j}\left(x_i\right)=\max_{p=\overline{1,l_j}}\left\{W_{jp}\min_{j=\overline{1,n}}\left[\mu^{a_i^{jp}}\left(x_i\right)\right]\right\}. \tag{4.33}$$

Let's randomly generate the rule weights W_{jp} of the knowledge base (KB) in the interval from 0 to 1.

Determining the chromosome efficiencies. The efficiency of each chromosome in the population for each set of weights W_{jp}, is determined using the objective function. Then, the chromosomes are sorted in ascending (or descending) order of their efficiency (forming a sequence based on error magnitude).

Selection. At this stage, chromosomes with the least efficiency are discarded if the total number of chromosomes in the population exceeds the permissible limit. Thus, the computational volume performed in the algorithm remains constant regardless of the iteration number.

Crossover. Two chromosomes are randomly selected from the population, considering their efficiency, and starting from a random position, they exchange genes. There can be multiple crossover points. If the two points defined by these chromosomes in the search space are in the vicinity of the same extremum, the average value between these points, resulting from the crossover, will be closer to the extremum. This is somewhat analogous to the gradient method. However, if the two points defined by these chromosomes are in the vicinity of different extrema, the average value between them will be random, akin to the random search method.

Mutation. This is fully analogous to the random search method. The values of individual genes in the population are randomly changed, i.e., the search point's position in the search space is altered.

End of the search. The search terminates if, over *L* L iterations, the efficiency of the best chromosome has increased by less than λ. Otherwise, a new iteration begins. A single iteration is called a generation. For instance, if the most efficient chromosome was found in the 30th generation, the condition for terminating the search algorithm was met at the 30th iteration, and the best chromosome in the final selection is considered the optimal solution.

Thus, the methodology for applying genetic algorithms to tune the information system for processing various types of data in the ACS is reduced to the following algorithm (**Fig. 4.6**):

1. Formation of reference states for the process of processing various types of data in the ACS.
2. Formation of the initial population.
3. Determination of the proximity criterion for tuning the fuzzy identifier.

4. Tuning the weights of the production rules in the knowledge base (KB) for each element of the ACS.

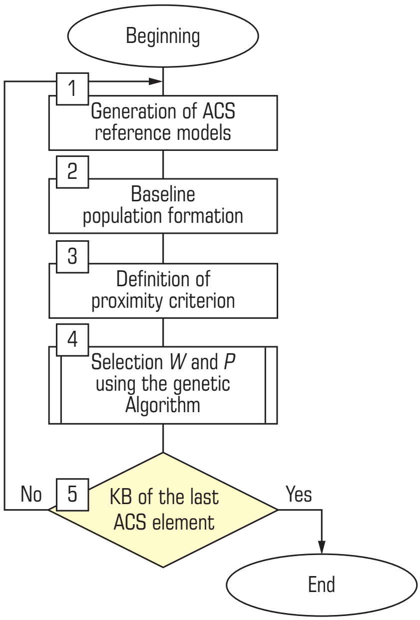

Fig. 4.6 Algorithm for applying the genetic algorithm (GA) to tune the information system for processing various types of data in the ACS

Formation of the initial population.

When calculating the rule weights in the knowledge base (KB), let's assume them to be equal to one.

Let's randomly generate the rule weights in the knowledge base (KB) within the interval from 0 to 1, i.e. $W_{jp} \in [0,1]$. Since there are 43 rules in the knowledge base (KB) and the number of chromosomes is 14, it is necessary to form a two-dimensional array of rule weight sets for tuning the information system for assessing the operational situation. The efficiency of each chromosome in the population for each set of weights is determined using the objective function W_{jp}, after that, the chromosomes are sorted in ascending (or descending) order of their efficiency.

Selection. At this stage, chromosomes with the lowest efficiency are discarded if the total number of chromosomes in the population exceeds the permissible limit. This means that the computational volume performed in the algorithm remains constant regardless of the iteration number.

Let's perform sorting of the chromosomes based on their efficiency:

$$\Delta_3, \Delta_9, \Delta_{12}, \Delta_{13}, \Delta_2, \Delta_4, \Delta_1, \Delta_{14}, \Delta_8, \Delta_7, \Delta_5, \Delta_{11}, \Delta_6, \Delta_{10}.$$

The highest efficiency, i.e., the smallest decision-making error, is held by the third and ninth chromosomes, while the lowest efficiency, i.e., the largest decision-making error, is held by the sixth and tenth chromosomes.

Crossover. In this case, for the crossover operation, chromosomes No. 3 and No. 9 are chosen as parent chromosomes. The resulting offspring chromosomes from the crossover are recorded in place of chromosomes No. 6 and No. 10, respectively.

Mutation. The mutation operation is performed on chromosome No. 10, and if the sum of the weights exceeds one, normalization is carried out: 0.15; 0.09; 0.08; 0.02; 0.2; 0.13; 0.07; 0.06; 0.16; 0.04.

Let's proceed to calculate its efficiency. The result obtained is $\Delta=1.1$, which is significantly better than the efficiency of the two previous chromosomes, the optimization of which we perform.

End of search.

CONCLUSIONS

1. Based on the analysis conducted in the study, it has been established that the application of fuzzy graphs and the mathematical apparatus of fuzzy logic in decision support tasks for data distribution and evaluating the process of processing various types of data under different conditions, including uncertainty, allows for the distribution of data between the elements of the ACS based on the importance of the ACS elements and the number of features in real-time.

2. The methodology for the rational distribution of data based on the importance of ACS elements and the number of features in the ACS under uncertainty conditions has been improved. This methodology differs from existing ones by combining the mathematical apparatus of information theory, fuzzy logic, and expert evaluation, which allowed for the formalization of features in a unified parameter space and, through the intellectualization of information processing processes, achieved more efficient data handling.

A quantitative assessment of the effectiveness of the proposed methodology has been conducted. The results of this assessment showed that the distribution of data between the elements of the ACS based on importance and the number of features using the proposed methodology increases the operational speed of data processing and decision-making regarding the state of the process of processing various types of data by 15–17 %.

3. The methodology for tuning the information system for evaluating the process of processing various types of data in the ACS under uncertainty conditions has been improved by using a genetic algorithm. In conditions where the application of other methods is limited due to the inability to vary

individual features with fixed values of other indicators, this approach has improved the operational speed of the developed information system for evaluating data processing in the ACS.

The scientific result is the improvement of the genetic algorithm for the differentiated tuning of the fuzzy knowledge base of the information system for evaluating the processing of various types of data in the ACS based on a posteriori data.

A quantitative evaluation of the effectiveness of the improved methodology has been carried out. The results of this evaluation showed that the use of the proposed methodology increases the operational efficiency of tuning the information system for processing various types of data in the ACS under uncertainty conditions.

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. 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

DOI: 10.15587/978-617-8360-13-9.CH5

Svitlana Kashkevich, Olena Shaposhnikova, Nataliia Saienko,
Viktoriia Shevchenko, Nataliia Holeho, Taras Zaloznyi

CHAPTER 5

INTELLECTUAL METHODS FOR EVALUATING THE STATE OF UAV CHANNELS

ABSTRACT

In this section of the research, intellectual methods for assessing the state of unmanned aerial vehicle (UAV) channels are proposed. During the study, the authors:

1. Developed an intellectual method for assessing the state of UAV channels based on the application of fuzzy set theory and artificial neural networks, which, while being sufficiently simple, allows for obtaining highly accurate solutions. The distinctive features of the proposed method include:

– simultaneous assessment of the UAV channel state based on multiple evaluation indicators (impulse response, frequency response, and bit error rate);

– real-time continuous assessment of several channel state characteristics;

– continuous assessment of multiple channel state characteristics in both the downlink and uplink channels;

– obtaining channel state assessments for each indicator on separate layers of the neural network through the construction of membership functions;

– following the assessment of individual channel characteristics by separate neural network layers, a generalized channel state assessment is formed at their output.

2. Proposed a methodology for identifying the state of UAV control and data transmission channels.

The novelty of the proposed methodology lies in:

– considering a corrective coefficient during calculations to account for the degree of uncertainty regarding the state of UAV channels;

– adding a corrective coefficient to address data noise resulting from distortions in information about UAV channel states;

– reducing computational costs when assessing the state of UAV channels;

– enabling calculations with input data of various natures and units of measurement.

KEYWORDS

Artificial intelligence, heterogeneous data processing, control and data transmission channels, reliability, efficiency.

5.1 DEVELOPMENT OF A METHOD FOR INTELLECTUAL ASSESSMENT OF UNMANNED AERIAL VEHICLE CHANNEL STATES

A generalized (comprehensive) assessment refers to obtaining an evaluation of the state of each UAV channel by a separate layer of the neural network based on frequency response, impulse response, and bit error rate [1–21]. After the layers of the neural network perform their evaluations, a generalized channel state assessment is formed at its output [22–30]. This generalized assessment enables the determination of mechanisms for adjusting UAV channel characteristics in terms of power level, frequency range, and conducting a quantitative evaluation of interference impact using the bit error rate [31–41].

Fig. 5.1 illustrates the principle of using an artificial neural network for assessing UAV channel state parameters.

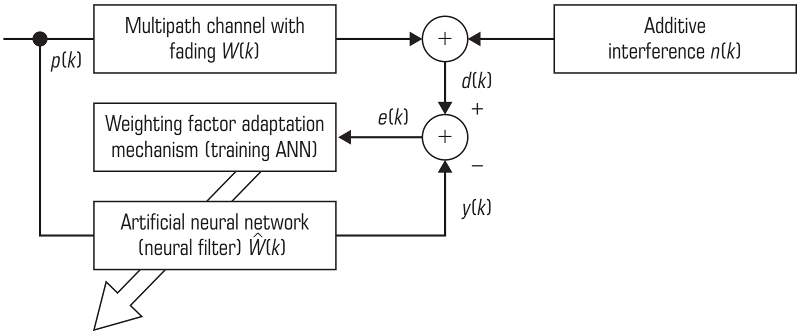

Fig. 5.1 Principle of adaptive assessment using an artificial neural network (ANN)

The method for intellectual assessment of UAV channel states, the implementation algorithm of which is shown in **Fig. 5.2**, consists of the following stages.

1. Input of initial data.

The parameters of the transmitting device and the communication channel are entered $\Psi = \{\psi_i\}$, $i = \overline{1,m}$, where $\psi_1 \ldots \psi_m$ – the signal ensemble positionality M, and the maximum power of the useful signal are specified $P_{c\ maks}$, threshold values for the signal-to-noise ratio (SNR) are specified Q^2_{thr}, the information transmission rate is specified v_i, parameters of the error-correcting code are specified (code rate R; the number of errors the code can correct s).

Constraints: modulation type – quadrature amplitude modulation (QAM), phase modulation (PM); signal ensemble dimensionality: $4 \le M \le 256$; type of error-correcting code – convolutional codes with a rate $R = 0.5–0.9$.

CHAPTER 5

2. Assessment of the UAV channel state.

Let's explain the procedure for assessing the channel state. The channel state assessment procedure involves obtaining a comprehensive evaluation of the UAV channel state. This procedure includes the parallel calculation of the channel's impulse response x_1, the frequency response of the channel state x_2 and the bit error rate x_3.

Thus, obtaining a comprehensive evaluation of the UAV channel state can be represented [1–9] as follows:

$$y = f\left(x_1, x_2, \ldots, x_n\right).$$

Artificial neural networks (ANNs), which are adaptive systems simulating the functionality of the nervous system of living organisms, are widely used for adaptive signal processing, modeling, analysis, and evaluation of the performance of UAV channels.

Adaptive filters using ANNs are capable of functioning effectively under conditions of a priori uncertainty about the properties of the external environment [10]. In the proposed method, each layer of the neural network evaluates an individual characteristic of the UAV channel state. Subsequently, the output of the neural network forms a generalized assessment of the UAV channel state.

3. Evaluation of the frequency response of the UAV channel.

Let the signal received at the analyzed frequency be represented as $A_s(t)$ and the additive concentrated interference $B_n(t)$ is a narrowband, quasi-stationary normal random process with a symmetric spectrum. Under these assumptions, these processes can be represented through their quadrature components [11]:

$$A_s(t) = Y_s(t)\cos\omega_s t + Y_n(t)\sin\omega_s t, \tag{5.1}$$

and

$$B_n(t) = X_s(t)\cos\omega_n t + X_n(t)\sin\omega_n t, \tag{5.2}$$

where $Y_s(t)$, $X_s(t)$ – in-phase; $Y_n(t)$, $X_n(t)$ – components represent the signal and interference, respectively; ω_s and ω_n – denote the mean frequencies of the signal and interference spectra. Going forward, it is possible to assume that $\omega_s \approx \omega_n = \omega_0$, with ω_0 precisely determined.

In the case of Rayleigh fading, which is most typical for channels with multipath propagation of radio waves, the quadrature components of the signal $Y_s(t)$, $Y_n(t)$ and the interference $X_s(t)$, $X_n(t)$ are pairwise independent normal Markov random processes with zero mean and variances $\sigma^2_{Y_s} = \sigma^2_{Y_n} = \sigma^2_s$, $\sigma^2_{X_s} = \sigma^2_{X_n} = \sigma^2_n$ [42–55].

CHAPTER 5

The correlation functions of the quadrature components in this case can be expressed as:

$$R_{Y_s}(\tau) = R_{Y_n}(\tau) = R_{Y_{qw}}(\tau) = \sigma_s^2 e^{-\alpha_s \tau}, \tag{5.3}$$

and

$$R_{X_s}(\tau) = R_{X_n}(\tau) = R_{X_{qw}}(\tau) = \sigma_n^2 e^{-\alpha_n \tau}, \tag{5.4}$$

where $\alpha_s = \frac{1}{\tau_s}$, $\alpha_n = \frac{1}{\tau_n}$ – the parameters of the correlation functions characterize the rates of change of the signal and interference in the quadrature channels of reception, respectively.

To obtain smoothed estimates of the signal and interference voltages over the evaluation interval ($\hat{y}(t) = 20\lg \hat{\hat{A}}_s(t)$ and $\hat{x}(t) = 20\lg \hat{\hat{B}}_n(t)$), the dependencies can be utilized through the corresponding estimates of the quadrature components smoothed over the same time interval:

$$\hat{y}(t) = 20\lg \sqrt{\hat{\hat{Y}}_s^2(t) + \hat{\hat{Y}}_n^2(t)}, \tag{5.5}$$

$$\hat{x}(t) = 20\lg \sqrt{\hat{\hat{X}}_s^2(t) + \hat{\hat{Y}}_n^2(t)}. \tag{5.6}$$

The estimate of the ratio of smoothed signal and interference voltages (levels) is determined by the following expression:

$$s(t) = y(t) - x(t). \tag{5.7}$$

The model that defines the variation of the smoothed components of the signal $\tilde{Y}_{qw}(t)$ and interference $\tilde{X}_{qw}(t)$ in each quadrature measurement channel in discrete time is described in a two-dimensional state space by the vector difference stochastic equation of the form:

$$\tilde{\boldsymbol{X}}(k+1) = \Phi(k+1)\tilde{\boldsymbol{X}}(k) + \Gamma(k+1)\tilde{\boldsymbol{U}}(k), \tag{5.8}$$

where $\tilde{\boldsymbol{X}}(k+1) = \left\| \tilde{Y}_{qw}(k+1), \tilde{X}_{qw}(k+1) \right\|^{\mathrm{T}}$ – the state vector of the signal and intentional interference components:

$$\Phi(k+1) = \mathrm{diag}\left\| e^{-\alpha_y \Delta t}, e^{-\alpha_x \Delta t} \right\|;$$

$$\Gamma(k+1) = \mathrm{diag}\left\| \sqrt{\frac{2}{\alpha_y}}(1 - e^{-\alpha_y \Delta t}); (1 - e^{-\alpha_x \Delta t}) \right\|;$$

CHAPTER 5

where $\tilde{\boldsymbol{U}}(k)=\left\|\tilde{U}_1(k),\tilde{U}_2(k)\right\|^{\mathrm{T}}$ – the vector of white Gaussian noise with zero mean and a covariance matrix:

$$Q(k)=\mathrm{diag}\left\|\frac{\alpha_y\sigma_y^2(1+e^{-\alpha_y\Delta t})}{2(1-e^{-\alpha_y\Delta t})};\ \frac{\alpha_x\sigma_x^2(1+e^{-\alpha_x\Delta t})}{2(1-e^{-\alpha_x\Delta t})}\right\|,$$

where σ_y^2 and σ_x^2 – variances; α_y and α_x – the parameters of the correlation functions of the smoothed quadrature components of the signal and interference, respectively.

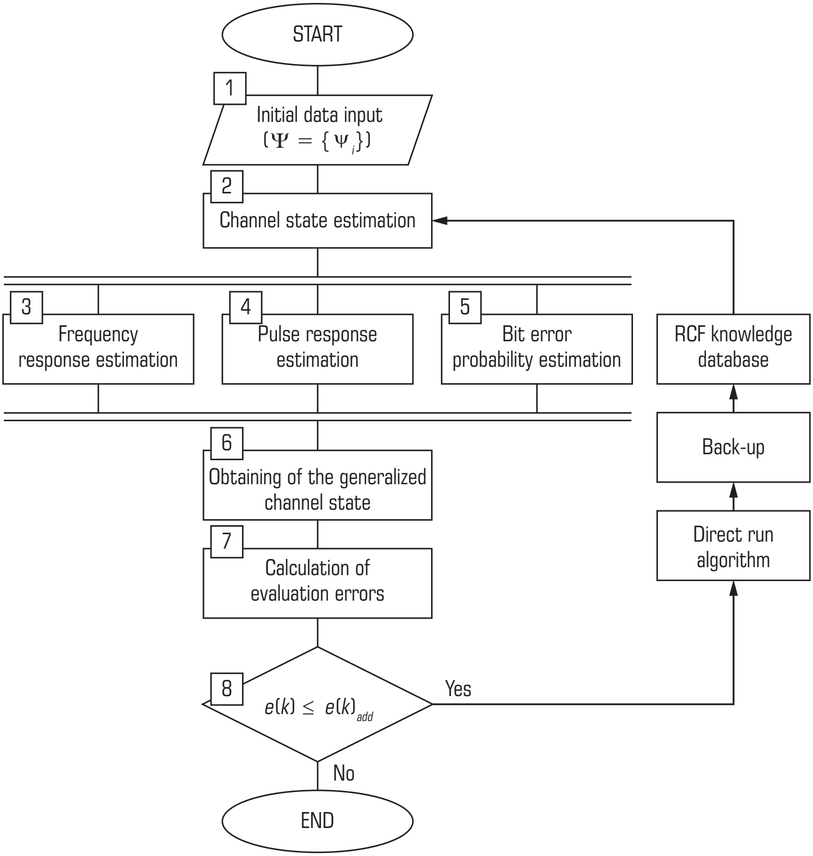

Fig. 5.2 Algorithm for implementing the intellectual assessment of UAV channel states

The signal and interference in each quadrature measurement channel are observed in an additive mixture with measurement noise:

$$\tilde{Z}_{qw}(k+1) = \boldsymbol{A}(k+1)\widetilde{\boldsymbol{X}}(k+1) + \tilde{N}_{qw}(k+1), \tag{5.9}$$

where $\boldsymbol{A}(k+1) = \|1, 1\|$, and $\tilde{N}_{qw}(k+1)$ – a Gaussian white sequence with zero mean and a covariance function:

$$R_n(k, l) = \sigma_n^2 \delta_{k,l}.$$

The solution to the optimal filtering problem for the state and observation model described by equations (5.8) and (5.9) leads to the following recursive computational algorithm [12]:

$$\widetilde{\boldsymbol{X}}(k+1) = \left\| e^{-\alpha_y \Delta t} \hat{\tilde{Y}}_{qw}(k), \quad e^{-\alpha_x \Delta t} \hat{\tilde{X}}_{qw}(k) \right\|^{\mathrm{T}} + \boldsymbol{K}(k+1) \times$$

$$\times \left\{ \tilde{Z}_{qw}(k+1) - \left[e^{-\alpha_y \Delta t} \hat{\tilde{Y}}_{qw}(k) + e^{-\alpha_x \Delta t} \hat{\tilde{X}}_{qw}(k) \right] \right\}. \tag{5.10}$$

In expression (5.10), the matrix weighting coefficient is determined as:

$$\boldsymbol{K}(k+1) = \left\| \begin{matrix} \dfrac{\rho_{11}(\Delta t) + \rho_{12}(\Delta t)}{\rho_{11}(\Delta t) + 2\rho_{12}(\Delta t) + \rho_{22}(\Delta t) + \sigma_n^2} \\ \dfrac{\rho_{22}(\Delta t) + \rho_{12}(\Delta t)}{\rho_{11}(\Delta t) + 2\rho_{12}(\Delta t) + \rho_{22}(\Delta t) + \sigma_n^2} \end{matrix} \right\|, \tag{5.11}$$

where

$$\rho_{11}(\Delta t) = P_{11}(k) e^{-2\alpha_y \Delta t} + \sigma_y^2 (1 - e^{-2\alpha_y \Delta t}),$$

$$\rho_{12}(\Delta t) = P_{12}(k) e^{-(\alpha_y + \alpha_x)\Delta t},$$

$$\rho_{22}(\Delta t) = P_{22}(k) e^{-2\alpha_x \Delta t} + \sigma_x^2 (1 - e^{-2\alpha_x \Delta t}). \tag{5.12}$$

In expressions (5.12) $P_{11}(k)$, $P_{22}(k)$, $P_{12}(k) = P_{21}(k)$ – elements of the mean-square error matrix of the estimation after k k-measurement steps.

The computational algorithm described by (5.10)–(5.12) can be utilized for analyzing the quality of UAV channels using specific test signals.

When monitoring the quality of UAV channels directly during the information transmission process, uncertainty arises regarding the presence or absence of signals in the reception channels.

One of the most constructive approaches to addressing this issue is the synthesis of computational control algorithms based on the combined application of signal detection procedures and parameter estimation for both the signals and concentrated interference.

Let the binary signals $A_s^{(l)}(t)\,(l=1,2)$ have identical energy and, like the additive interference $B_n(t)$ be narrowband, quasi-stationary Markov random processes with symmetric spectra. Frequency-modulated signals $A_s^{(l)}(t)\,(l=1,2)$ can be represented through their quadrature components:

$$A_s^{(l)}(t)=Y_s(t)\cos\omega_{0l}t+Y_n(t)\sin\omega_{0l}t. \tag{5.13}$$

Without loss of generality, let's consider the case of symmetric interference of concentrated noise on the signals:

$$B_n(t)=X_n(t)\cos\omega_{0l}t+X_n(t)\sin\omega_{0l}t. \tag{5.14}$$

For Rician and Rayleigh radio communication channels, the dynamics of the quadrature components of the signal and interference at discrete time moments t_k $(k=0,1,2,\ldots)$ can, by analogy with expression (5.8), be represented by systems of stochastic vector difference equations:

$$\bar{\boldsymbol{X}}_{s(n)}(k+1)=\Phi(k+1)\bar{\boldsymbol{X}}_{s(n)}(k)+\Gamma(k+1)\bar{\boldsymbol{U}}(k), \tag{5.15}$$

where

$$\bar{\boldsymbol{X}}_s(k+1)=\left\|\bar{Y}_s(k+1),\bar{X}_s(k+1)\right\|^{\mathrm{T}};$$

$$\bar{\boldsymbol{X}}_n(k+1)=\left\|\bar{Y}_n(k+1),\bar{X}_n(k+1)\right\|^{\mathrm{T}};$$

$$\bar{\boldsymbol{U}}(k)=\left\|\bar{U}_1(k),\bar{U}_2(k)\right\|^{\mathrm{T}}.$$

The matrices $\Phi(k+1)$, $\Gamma(k+1)$ and $Q(k)$ however, instead of the parameters α_y, α_x and σ_y^2, σ_x^2 are used, respectively α_s, α_n and σ_s^2, σ_n^2.

It should be noted that the time discretization interval:

$$\Delta t=t_{k+1}-t_k\ (k=0,1,2\ldots),$$

is chosen under the condition $\Delta t << \min(\tau_s,\tau_n)$. According to statistical data, the values of the correlation intervals for the amplitudes of fading signals (interference) in UAV channels range from tenths of a second to several seconds [56–64].

CHAPTER 5

The scalar observation equations for signals and interference in the presence of noise can be written as follows:

$$\bar{z}(k+1)=\chi\left[\boldsymbol{A}\bar{\boldsymbol{X}}_s(k+1)\cos\omega_{01}k\Delta t+\boldsymbol{A}\bar{\boldsymbol{X}}_s(k+1)\sin\omega_{01}k\Delta t\right]+$$
$$+(1-\chi)\left[\boldsymbol{A}\bar{\boldsymbol{X}}_s(k+1)\cos\omega_{02}k\Delta t+\boldsymbol{A}\bar{\boldsymbol{X}}_s(k+1)\sin\omega_{02}k\Delta t\right]+\bar{N}(k+1), \quad (5.16)$$

where A = | |I, 1| |; χ – a random variable such that:

$$\chi=\begin{cases}1 & \text{with probability } P;\\ 0 & \text{with probability } Q=1-P,\end{cases}$$

where P, Q – the a priori probabilities of signal transmission $U_s^{(l)}(t)$ (in particular, $P = Q = 0.5$); $\bar{N}(k+1)$ – normal white noise with zero mean and a covariance function $R_n(k,l)=\sigma_n^2\delta_{k,l}$.

The optimal decision rule, according to the ideal observer criterion for a priori equally probable transmitted signals, at time $T_a = n\Delta t$ is defined by the inequality:

$$\Lambda\left[\bar{z}(n)\right] \underset{<}{>} 0, \quad (5.17)$$

where the logarithm of the likelihood ratio is expressed in sequential form as:

$$\Lambda\left[\bar{z}(k+1)\right]=\ln\left\{\frac{w\left[\bar{z}(k+1)/\chi=1\right]}{w\left[\bar{z}(k+1)/\chi=0\right]}\right\}=\Lambda\left[\bar{z}(k)\right]+\ln\left\{\frac{w\left[\bar{z}(k+1)/\bar{z}(k),\chi=1\right]}{w\left[\bar{z}(k+1)/\bar{z}(k),\chi=0\right]}\right\}, \quad (5.18)$$

where $\Lambda\left[\bar{z}(0)\right]=0$.

The posterior probability densities $w\left[\bar{z}(k+1)/\bar{z}(k),\chi\right]$ in expression (5.18) are Gaussian and, in this case, take the form:

$$w\left[\bar{z}(k+1)/\bar{z}(k),\chi\right]=\left\{2\pi\det\left[\boldsymbol{AP}(k+1/k)\boldsymbol{A}^{\mathrm{T}}+\sigma_n^2\right]\right\}^{\frac{1}{2}}\times$$
$$\times\exp\left\{-\frac{\left\{\bar{z}(k+1)-\boldsymbol{A}\Phi(k+1)[\hat{\hat{\mathrm{X}}}_s(k)\cos\omega_{0l}k\Delta t+\hat{\hat{\mathrm{X}}}_s(k)\sin\omega_{0l}k\Delta t]\right\}^2}{2\pi[\boldsymbol{AP}(k+1/k)\boldsymbol{A}^{\mathrm{T}}+\sigma_n^2]}\right\}, \quad (5.19)$$

where $\bar{\boldsymbol{X}}_s(k)=\left\|\hat{Y}_s(k),\hat{Y}_s(k)\right\|^{\mathrm{T}}$; $\bar{\boldsymbol{X}}_n(k)=\left\|\hat{Y}_n(k),X_n(k)\right\|^{\mathrm{T}}$ – the vectors of estimates of the quadrature components of the signal and concentrated interference, calculated similarly to the algorithm in (5.15); $\mathbf{P}(k+1/k)$ – the transition matrix of estimation errors.

Based on expressions (5.17)–(5.19), let's arrive at the following recursive computational algorithm for joint detection and estimation:

$$\sum_{k=0}^{n-1}\left\{\bar{z}(k+1)-\boldsymbol{A}\Phi(k+1)[\hat{\bar{\mathbf{X}}}_s(k)\cos\omega_{01}k\Delta t+\hat{\bar{\mathbf{X}}}_n(k)\sin\omega_{01}k\Delta t]\right\}^2 \gtrless$$

$$\gtrless \sum_{k=0}^{n-1}\left\{\bar{z}(k+1)-\boldsymbol{A}\Phi(k+1)[\hat{\bar{\mathbf{X}}}_s(k)\cos\omega_{02}k\Delta t+\hat{\bar{\mathbf{X}}}_n(k)\sin\omega_{02}k\Delta t]\right\}^2. \tag{5.20}$$

Under conditions of interference structurally similar to the signal, it is necessary to perform separate real-time estimation of the amplitudes of the signal and interference without prior decisions on the reception of informational symbols. For this, information about the time autocorrelation functions of the modulated signal and interference at the output of UAV channels is required.

Such an estimation establishes the degree of influence of the manipulation speed ratio of the signal and interference, as well as the rate of change in the characteristics of the propagation environment, on the radio channel quality. This enables recommendations to be made for selecting optimal operating frequencies for UAV channels, considering the correlation properties of the channels, signals, and interference.

4. Evaluation of the impulse response of UAV channels.

The estimation of impulse responses at the *k*-th step for UAV channels will be represented by the weight coefficient matrix of the Focused Neural Network Filter (FNNF) in the form:

$$H_{MIMO}(k)=\begin{bmatrix} W_l^{11}(k) & W_l^{12}(k) & \cdots & W_l^{1N}(k) \\ W_l^{21}(k) & W_l^{22}(k) & \cdots & W_l^{2N}(k) \\ \cdots & \cdots & \cdots & \cdots \\ W_l^{N1}(k) & W_l^{N2}(k) & \cdots & W_l^{NN}(k) \end{bmatrix}, \tag{5.21}$$

where $W_l^{NN}=\left[w_0^{NN},w_1^{NN},\ldots,w_{L-1}^{NN}\right]^{\mathrm{T}}$ – the column vector represents the samples of the impulse response in the direction from the M_t-th transmitter to the M_r-th receiver of the UAV.

5. Evaluation of the bit error rate (BER) for UAV channels.

The measurement results for the bit error rate of a specific signal type are obtained according to the mathematical relationships derived in [65–71].

6. Obtaining a generalized assessment of the UAV channel state.

During the channel state evaluations (**Steps 3–5**), it becomes necessary to form a generalized assessment of the UAV channel state.

The primary tool of fuzzy logic, which allows transforming expert knowledge in the form of "IF-THEN" rules into mathematical models, is the membership function [72–80].

For this task, the membership function characterizes the degree of confidence of an expert that a certain value belongs to a fuzzy concept (term). Methods of fuzzy logical inference enable linking the membership functions of indicators and the signal-interference environment in the presence of a channel model expressed through "IF-THEN" rules.

$$a_i^p = \int_{\underline{x}_i}^{\overline{x}_i} \mu^{a_i^p}\left(x_i\right)/x_i, \tag{5.22}$$

$$d_j = \int_{\underline{d}}^{\overline{d}} \mu^{d_j}\left(d\right)/d. \tag{5.23}$$

Thus, the result of the UAV channel state assessment can be represented as [80–86]:

$$y = f\left(x_1, x_2, \ldots, x_n\right), \tag{5.24}$$

where $x_1, x_2, \ldots, x_n$ – the set of input parameter values of the UAV channel state, and y is the result of the UAV channel state assessment.

The ranges of the input parameters $x_i \in [\underline{x}, \overline{x}]$, $i = \overline{1,n}$ and the output value $y \in [\underline{y}, \overline{y}]$ or the UAV channel state evaluation is assumed to be known. Here $x_i\left(\overline{x}_i\right)$ – denotes the lower (upper) limits of the input parameters, and x_i, $i = \overline{1,n}$, $\underline{y}\left(\overline{y}\right)$ – represents the lower (upper) limits of the identification result.

Let $X^* = \left\langle x_1^*, x_2^*, \ldots, x_n^*\right\rangle$ – be the vector of fixed values of the input parameters of the UAV channel state, where $x_i^* \in \left[\underline{x}_i, \overline{x}_i\right]$, $i = \overline{1,n}$. The decision-making task is to determine the result x^* of the UAV channel state assessment based on the information about the input parameter vector $y^* \in Y$. A necessary condition for the formal solution of this task is the existence of the dependence (5.24). To establish this dependence, the signal-interference environment and the output decision on the generalized channel state assessment must be represented as linguistic variables defined on universal sets [9, 10, 12, 14–18]:

$$X_i = \left[\underline{x}_i, \overline{x}_i\right], \tag{5.25}$$

$$Y = \left[\underline{y}, \overline{y}\right]. \tag{5.26}$$

To evaluate such linguistic variables, it is proposed to use qualitative terms that constitute a term set [9, 10, 12, 14–18]:

$A_i = \left\{a_i^1, a_i^2, \ldots, a_i^{g_i}\right\}$ – linguistic term of the variable x_i, $i = \overline{1,n}$, $D = \left\{d_1, d_2, \ldots, d_m\right\}$ – linguistic term of the variable y, where a_i^p – p-th linguistic term of the variable x_i, $p = \overline{1, g_i}$, $i = \overline{1,n}$; d_j – j-th linguistic term of the variable y; m – the number of different decisions. The cardinalities of the term sets A_i, $i = \overline{1,n}$ can vary, i.e. $\left(g_1 \neq g_2 \neq \ldots \neq g_n\right)$. The names of terms $a_i^1, a_i^2, \ldots, a_i^{k_i}$ may differ for different linguistic variables x_i, $i = \overline{1,n}$.

The linguistic terms $a_i^p \in A_i$ and $a_i^p \in A_i$ i $d_j \in D$, $p = \overline{1,k_i}$, $i = \overline{1,n}$, $j = \overline{1,m}$ must be considered as fuzzy sets defined on the universal sets X_i, Y (5.25), (5.26).

The fuzzy sets a_i^p and d_j are determined by the relationships [9, 10, 12, 14–18]:

$$a_i^p = \int_{\underline{x}_i}^{\overline{x}_i} \mu^{a_i^p}(x_i) / x_i, \tag{5.27}$$

$$d_j = \int_{\underline{d}}^{\overline{d}} \mu^{d_j}(d) / d, \tag{5.28}$$

where $\mu^{a_i^p}(x_i)$ – the membership function of the variable $x_i \in [\underline{x}, \overline{x}]$, $i = \overline{1,n}$ to the term $a_i^p \in A_i$, $p = \overline{1,k_i}$; μ^{d_j} – and the membership function of the variable $y \in [\underline{y}, \overline{y}]$ to the term-decision $d_j \in D$, $j = \overline{1,m}$. In relationships (5.27) and (5.28), the integral symbols indicate the aggregation of pairs $\mu(u) / u$.

Let N – the amount of expert survey data linking the input indicators and the output assessment of the UAV channel state is distributed as follows:

$$N = g_1 + g_2 + \ldots + g_m,$$

where g_j – the number of expert data points corresponding to an output decision is $d_j \in D$, $j = \overline{1,m}$, m – the number of output decisions is $g_1 \neq g_2 \neq \ldots \neq g_m$.

The number of selected expert data points is less than the total possible combinations of changes in the input indicators of the UAV channel state.

After numbering, the known expert data about the channel state can be presented in the form of a knowledge matrix [9, 10, 12, 14–18] (**Table 5.1**).

The knowledge matrix is formed according to the following rules:

– the dimension of such a matrix is $(n+1) \times N$, where $(n+1)$ – the number of rows, $N = g_1 + g_2 + \ldots + g_m$ – the number of columns;

– the first n columns correspond to the input indicators of the UAV channel state x_i, $i = \overline{1,n}$ and $(n + 1)$-th column corresponds to the values d_j the output decision y, $j = \overline{1,m}$;

– each row of the matrix represents a specific combination of values of the input indicators of the UAV channel state, assigned by an expert to one of the possible channel state values d_j the first g_j rows correspond to the value d_1, while the last g_m rows correspond to the value d_m;

– the element a_i^{jp}, located at the intersection of the i-th column and jp-th row, corresponds to the linguistic assessment of the indicator x_i in the row of the fuzzy knowledge base with the number jp, the linguistic assessment a_i^{jp} is selected from the term – set of the corresponding indicator x_i i.e. $(a_i^{jp} \in A_i, i = \overline{1,n}, j = \overline{1,m}, p = \overline{1,l_j})$.

Thus, expression (5.24), which establishes the relationship between the input indicators x_i and the output assessment y, is formalized into a system of logical statements (5.30), based on the fuzzy knowledge base represented by the knowledge matrix (**Table 5.1**).

Graphically, the process of generalized assessment of the UAV channel can be depicted as shown in **Fig. 5.3**.

Table 5.1 Fuzzy knowledge matrix of UAV channels

The number of the input combination of values	Input variables UAV channels				Output variables
	x_1	x_2	x_i	x_n	Y
11	a_1^{11}	a_2^{11}	… a_i^{11} …	a_n^{11}	d_1
12	a_1^{12}	a_2^{12}	… a_i^{12} …	a_n^{12}	
…	…	…	…	…	
lg_1	$a_1^{1g_1}$	$a_2^{1g_1}$	… $a_i^{1g_1}$ …	$a_n^{1g_1}$	
j1	a_1^{j1}	a_2^{j1}	… a_i^{j1} …	a_n^{j1}	d_i
j2	a_1^{j2}	a_2^{j2}	… a_i^{j2} …	a_n^{j2}	
…	…	…	…	…	
jg_j	$a_1^{jg_j}$	$a_2^{jg_j}$	… $a_i^{jg_j}$ …	$a_n^{jg_j}$	
…	**…**	**…**	**…**	**…**	**…**
m1	a_1^{m1}	a_2^{m1}	… a_i^{m1} …	a_n^{m1}	d_m
m2	a_1^{m2}	a_2^{m2}	… a_i^{m2} …	a_n^{m2}	
…	…	…	…	…	
mg_m	$a_1^{mg_m}$	$a_2^{mg_m}$	… $a_i^{mg_m}$ …	$a_n^{mg_m}$	

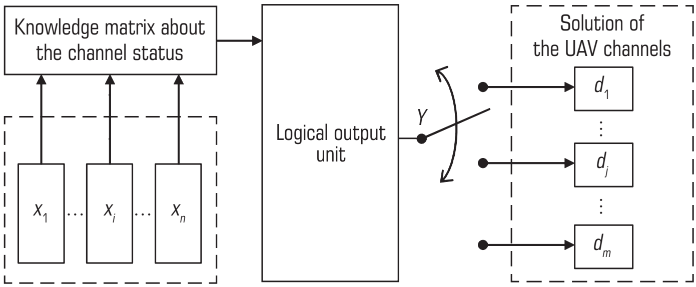

Fig. 5.3 Formation of the generalized assessment of the channel state

From the analysis of the UAV channel functioning under various signal conditions, let's determine the directions for assessing the quality of the UAV channel.

These include the similarity of indicators characterizing the quality of the UAV channel and changes during the communication session until the decision on the channel quality is made.

The formation of the generalized assessment of the UAV channel state can be written as an equation:

$$D(k)=f\begin{bmatrix} Y_1(k-1),\ldots,Y_n(k-1), \\ Z_1(k-1),\ldots,Z_n(k-1) \end{bmatrix},$$

where $Y_1(k-1)$ – a vector characterizing the first indicator of the UAV channel quality assessment at $k-1$ the assessment step; $Y_n(k-1)$ – a vector characterizing the n-th indicator of the UAV channel quality assessment at $k-1$ the modeling step; $Z_1(k-1), \ldots, Z_n(k-1)$ – vectors characterizing the generalized assessment of the channel for each of the quality evaluation indicators.

In turn, the vector of communication channel quality assessment is determined by the following indicators:

$$Y_1, \ldots, Y_n, Z_1, \ldots, Z_n = \{k_{11}(x), \ldots, k_n(x)\}.$$

The possible states of the signal environment in the channel are defined by the set $d \in \{d_1, d_2, d_3\}$, where d_1 – the channel meets the standard (corresponds to maximum frequency efficiency); d_2 – individual quality indicators of the channel exceed the standard and require adjustment; d_3 – the channel is unsuitable for operation. The task of evaluation is to match each combination of signal environment indicators to one of the decisions d_i, $i=\overline{1,3}$. The indicator $g_{11}, \ldots, g_{21}, \ldots, g_{141}, \ldots, g_{145}$ will be considered as linguistic variables [12].

The formation of the generalized assessment of communication channel quality can be represented as a multi-level hierarchical tree of logical inference corresponding to the following states:

$$d = f_d\,(Z_1 \ldots Z_n), \tag{5.29}$$

$$Z = f_z\,(Y_1 \ldots Y_n), \tag{5.30}$$

$$Y_n = f_{y_n}\left(g_{n1}(x), g_{n2}(x), g_{n3}(x), g_{n4}(x), g_{n5}(x)\right). \tag{5.31}$$

For indicators with a quantitative representation, the range of variation is divided into four quanta. This enables the transformation of the continuous universal $U=[\underline{u},\overline{u}]$ into a discrete five-element set [12]:

$$\mathrm{U} = \{u_1, u_2, \ldots, u_5\},$$

where

$$u_1=\underline{u},\ u_2=\underline{u}+\Delta_1,\ u_3=u_2+\Delta_2,\ u_4=u_3+\Delta_3,\ u_5=u_4+\Delta_4,$$

where $\Delta_1+\Delta_2+\Delta_3+\Delta_4=\bar{u}-\underline{u}$, $\bar{u}\left(\underline{u}\right)$ – the lower and upper bounds of the indicator's range of variation, respectively. Then, all pairwise comparison matrices will have a dimension of 5×5. The choice of four quanta is determined by the ability to approximate nonlinear curves using five points [12].

To evaluate the values of linguistic variables $g_{11}, \ldots, g_{21}, \ldots, g_{141}, \ldots, g_{145}$ let's use a qualitative term scale. For evaluating linguistic variables D, Z_n, Y_n it is possible to use the following term sets: D, $Z_1 \ldots Z_n$, $Y_1 \ldots Y_n$ = {channel parameters meet the standard, some channel parameters exceed the standard and require adjustment, channel is unsuitable for operation}.

Each of the introduced terms represents a fuzzy set, defined by the corresponding membership function. In general, the input variables $x_1, x_2, \ldots, x_n$ can be specified by a number, a linguistic term, or based on the thermometer principle [12].

The evaluation of the UAV channel is performed using fuzzy logical equations [12], which constitute a knowledge matrix and a system of logical statements. These equations allow the computation of membership function values for different assessment results with fixed input indicator values. The decision with the highest membership function value is proposed as the result of the evaluation process [12].

The linguistic assessments α_i^{jp} of the variables $x_1, x_2, \ldots, x_n$ which are part of the logical statements regarding the decisions d_j, $j=\overline{1,m}$ (5.27), (5.28), are considered as fuzzy sets defined on universal sets:

$$X_i=\left[\underline{x}_i,\bar{x}_i\right],\ i=\overline{1,n}.$$

Let $\mu^{a_i^{jp}}\left(x_i\right)$ – be the membership function of the indicator $x_i \in \left[\underline{x},\bar{x}\right]$ to the fuzzy term $i=\overline{1,n}$, $j=\overline{1,m}$, $p=\overline{1,l_i}$; $\mu^{d_j}\left(x_1,x_2,\ldots,x_n\right)$ – be the membership function of the input variable vector $X = (x_1, x_2, \ldots, x_n)$ to the output assessment $y = d_j$, $j=\overline{1,m}$. The relationship between these functions is determined by the fuzzy knowledge base and can be represented as the following logical equations:

$$\begin{aligned}
&\mu^{d_j}\left(x_1,x_2,\ldots,x_n\right)=\mu^{a_1^{j1}}\left(x_1\right)\wedge\mu^{a_2^{j1}}\left(x_2\right)\wedge\ldots\wedge\mu^{a_n^{j1}}\vee\\
&\vee\mu^{a_1^{j2}}\left(x_1\right)\wedge\mu^{a_2^{j2}}\left(x_2\right)\wedge\ldots\wedge\mu^{a_n^{j2}}\left(x_n\right)\ldots\\
&\ldots\mu^{a_1^{jl_j}}\left(x_1\right)\wedge\mu^{a_2^{jl_j}}\left(x_2\right)\wedge\ldots\wedge\mu^{a_n^{jl_j}}\left(x_n\right),j=\overline{1,m}.
\end{aligned}\tag{5.32}$$

These equations are derived from the fuzzy knowledge base by substituting variables (linguistic terms) with membership functions and replacing the operations AND and OR with the operations $\wedge$ and $\vee$.

The system (5.32) can be written compactly as:

$$\mu^{d_j}\left(x_i\right)=\bigvee_{p=1}^{l_j}\left[\bigwedge_{i=1}^{n}\mu^{a_i^{jp}}\left(x_i\right)\right],\ j=\overline{1,m}.\tag{5.33}$$

Fuzzy logical equations are an analog of the fuzzy logical inference procedure introduced by Zadeh [9, 10, 12, 14–18], which is performed using the operations of "fuzzy (min-max) composition", where the operations AND and OR correspond to the min and max operations, respectively. From (5.33), obtain:

$$\mu^{d_j}(x_i) = \max_{p=\overline{1,l_j}} \left\{ \min_{j=\overline{1,n}} \left[\mu^{a_i^{jp}}(x_i) \right] \right\}. \quad (5.34)$$

From expression (5.34), it is evident that calculating the membership function requires only the membership functions of the variables to the fuzzy terms. Let's consider the procedure for calculating the membership function used in this method.

When selecting indirect methods for constructing the membership function, which transform expert information into a form convenient for use in UAV channel assessment, it is necessary to consider the computational complexity of implementation. For instance, constructing a membership function based on the pairwise comparison method requires forming a pairwise comparison matrix and solving the characteristic equations of this matrix to determine its eigenvector. However, this method has high computational complexity. Considering the possibility of calculating the membership function using ranking assessments, which are relatively easy to obtain through expert surveys, let's use the pairwise comparison method to calculate the membership function.

The algorithm for calculating the membership function of the UAV channel state includes the following Steps:

1. Select the quality indicator of the UAV channel to be assessed x_j, $j=\overline{1,m}$.
2. Set of fuzzy terms is specified $\{u_1, u_2, \ldots, u_l\}$, which are used for evaluating x.
3. For each term u_i, $i=\overline{1,l}$ a pairwise comparison matrix is formed:

$$T = \begin{vmatrix} 1 & \frac{r_2}{r_1} & \frac{r_3}{r_1} & \ldots & \frac{r_n}{r_1} \\ \frac{r_1}{r_2} & 1 & \frac{r_3}{r_2} & \ldots & \frac{r_n}{r_2} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \frac{r_1}{r_n} & \frac{r_2}{r_n} & \frac{r_3}{r_n} & \ldots & 1 \end{vmatrix}, \quad (5.35)$$

where $r_s(u_i)$ – the rank of element $u_i \in U$, which characterizes the significance of this element when forming the properties described by the corresponding fuzzy term $\tilde{S}$.

The matrix (5.35) has the following properties:

– the elements of the main diagonal are equal to 1 ($t_{ij} = 1, i=\overline{1,n}$);

– relative to the main diagonal, the elements are related by the expression $t_{ij} = 1/t_{ji}$;

– the condition of transitivity holds: $t_{ig}t_{gj} = t_{ij}$, since $\frac{r_i}{r_g}\frac{r_g}{r_j} = \frac{r_i}{r_j}$.

Thanks to these properties, the elements of other rows in the matrix T can easily be found from the known elements of one row. If the elements t_{gj}, g, $j=\overline{1,n}$ are known, any element t_{ij} can be found as: $t_{ij} = t_{gj}/t_{gi}$, $i,j,g=\overline{1,n}$.

Since matrix (5.35) can be interpreted as a pairwise comparison matrix of ranks, a twelve-point Saaty scale [18] can be used for expert evaluation of the elements of this matrix.

For the case:

$$t_{ij} = r_i / r_j = \begin{cases} 1 - \text{no advantage of } r_i \text{ over } r_j; \\ 3 - \text{ small advantage of } r_i \text{ over } r_j; \\ 5 - \text{larger advantage of } r_i \text{ over } r_j; \\ 7 - \text{substantial advantage of } r_i \text{ over } r_j; \\ 9 - \text{clear cut advantage of } r_i \text{ over } r_j; \\ 11 - \text{absolute advantage of } r_i \text{ over } r_j; \\ 2,4,6,8,10 - \text{interim comparison evaluation.} \end{cases}$$

4. Membership functions are determined as follows:

1) based on absolute rank assessments r_i, $i = \overline{1,n}$, which can be defined using a nine-point scale (1 – lowest rank, 9 – highest rank);

2) based on relative rank assessments $r_i / r_j = t_{ij}$, $i, j = \overline{1,n}$, which are determined by the pairwise comparison matrix (5.35), the membership function is calculated for each term.

The normalization of the obtained membership functions is performed by dividing by the highest membership degrees.

These relationships enable the calculation of the membership function using rank assessments, which are relatively easy to obtain when using neuro-fuzzy networks.

Using the knowledge matrix, the expert information about the UAV channel state can be represented as a system of fuzzy logical statements (5.35), which link the values of the input indicators x_i with one of the possible decisions d_j, $j = \overline{1,m}$:

$$\begin{aligned} & IF\left(x_1 = \alpha_1^{11}\right) AND \left(x_2 = \alpha_2^{11}\right) AND \ldots AND \left(x_n = \alpha_n^{11}\right) OR \left(x_1 = \alpha_1^{12}\right) \\ & \left(x_2 = \alpha_2^{12}\right) AND \ldots AND \left(x_n = \alpha_n^{12}\right) OR \ldots OR \left(x_1 = \alpha_1^{1l_1}\right) AND \left(x_2 = \alpha_2^{1l_1}\right) \\ & AND \ldots AND \left(x_n = \alpha_n^{1l_1}\right), THEN \;\; y = d_m, \end{aligned} \tag{5.36}$$

where $d_j \left(j = \overline{1,m}\right)$ – the linguistic assessment of the output variable y, which is defined from the term set D; a_i^{jp} – the linguistic assessment of the input indicator x_i in the p-th row j-th disjunction, chosen from the term set A_i, $\left(i = \overline{1,n}, j = \overline{1,m}, p = \overline{1,k_j}\right)$; g_j – the number of rules that define the output variable's value $y = d_j$.

Let's call the system (5.36) a fuzzy knowledge base, which is used to form a set of assessments for each of the indicators of the channel state evaluation [18, 19].

The fuzzy logical relations (5.36), along with the membership function of the fuzzy terms, allow the evaluation of the UAV channel state through the following algorithm:

1. The values of the quality indicators of the UAV channel are fixed according to the predefined criteria $X^* = \left(x_1^*, x_2^*, \ldots, x_T^*\right)$.

2. Using the membership function calculation algorithm, the membership function $\mu^j\left(x_i^*\right)$ is determined for the fixed values of the indicators x_i^*, $i = \overline{1,m}$.

3. Using the logical equations (5.36), let's calculate the membership function $\mu^{d_j}\left(x_1^*, x_2^*, \ldots, x_m^*\right)$ for the state vector $X^* = \left(x_1^*, x_2^*, \ldots, x_T^*\right)$ for all states d_1, d_2, …, d_n. In this case, the logical operations AND ($\wedge$) and *OR* ($\vee$) over the membership functions are replaced with the operations min and max [9, 10, 12, 14–18].

4. The decision d_j^* is determined for which:

$$d_j = \underset{j=\overline{1,m}}{\arg\max}\left(\mu d_j\left(x_i\right)\right).$$

The data in the matrices form a fuzzy knowledge base for assessing the quality of the UAV channel [17].

Using the tables and the AND and OR operations, a system of logical equations is written, linking the membership functions of the quality assessment decisions for the UAV channel with the membership functions of destabilizing factors. Thus, knowing the values of the membership functions for fuzzy terms, the quality of the UAV channel can be evaluated by solving the logical equations described above. The result of applying the proposed hierarchical system of fuzzy logical equations is the degree of membership of the UAV channel's state.

The output result of the evaluation and the quality indicators of the UAV channel are presented as linguistic variables, which are evaluated using the provided methods.

7. Calculation of the estimation error and training of the artificial neural network (ANN).

The principle of intellectual parameter estimation for the UAV channel using an ANN is shown in **Fig. 5.2**. It involves inputting a known training (learning) discrete sequence $p(k)$, into the ANN, which is processed in the neural network filter (ANN). As a result, the output signal $y(k)$ is obtained. This output signal is compared with the signal (sample) $d(k)$, received on the UAV channel's receiving side. The difference between them forms the error signal $e(k)$ (filter mismatch).

To correct the estimated values of the weight vector coefficients – the taps of the delay line, it is possible to use the following recursion:

$$\widehat{W}(k+1) = \widehat{W}(k) - \mu \nabla J\left(\widehat{W}(k)\right) = \widehat{W}(k) + 2\mu P^T(k)e(k), \tag{5.37}$$

where μ – the learning rate parameter of the neural network filter (NNF),

$$\nabla J\left(\widehat{W}(k)\right) = \frac{\partial e(k)}{\partial \widehat{w}(k)} = \frac{\partial e(k)}{\partial d(k)}\frac{\partial d(k)}{\partial \widehat{w}(k)} = -2P^T(k)d(k) + 2P^T(k)y(k) =$$

$$= -2P^T(k)\left(d(k) - y(k)\right) = -2P^T(k)e(k). \tag{5.38}$$

In the theory of artificial neural networks (ANNs), there are various methods for selecting the value of the learning rate parameter μ, with its value either remaining constant during the learning process or being adjusted adaptively. Fixing the value of the learning rate is considered the simplest form of determination, but it has many drawbacks and is now relatively rarely used. However, this method remains the most effective during the training of the neural network filter (NNF). The constant value of the learning rate parameter is set within the range $(0 < \mu < 1)$.

The task of the adaptive neural network filter is to minimize the error signal $e(k)$. To achieve this, an adaptation mechanism (tuning) of the weight coefficients of the neural network filter is used based on the analyzed error signal $e(k)$.

The adaptation (tuning) procedure involves searching for unknown parameters that ensure the adequacy of the neural network. The back-propagation method [9, 17] was used for training the neural network. Each iteration of the training procedure consists of two stages – the forward and backward passes.

The first stage of the training procedure (forward pass algorithm) follows the sequence of Steps:

1. Calculate the total weighted input signal $p_j(k)$ or each neuron of the current layer:

$$p_j\left(k\right)=\sum_{i=1}^{N_{1-1}} v_i w_{ij}. \tag{5.39}$$

2. Calculate the output signal $y_j(k)$ of each neuron of the current layer:

$$y_j\left(k\right)=\frac{1}{1+e^{-kp_j(k)}}. \tag{5.40}$$

3. If the current layer is not the output layer, move to the next layer and repeat the procedure from **Step 1**.

4. Calculate the error $e(k)$ of the neural network:

$$e\left(k\right)=\frac{1}{2}\sum_{j=1}^{N_K}\left(y_j\left(k\right)-\widehat{d}_j\left(k\right)\right)^2, \tag{5.41}$$

where $\widehat{d}_j\left(k\right)$ – the reference output value of the j-th neuron of the output layer, where N_K – the number of neurons in the output layer.

The second stage of the training procedure (backward pass algorithm) follows the sequence of Steps:

1. Determine the rate of change of the error with respect to the output signal for each neuron in the output layer (EA):

$$EA_j^K=\frac{\partial e\left(k\right)}{\partial y_j\left(k\right)}=\left(\widehat{d}_j\left(k\right)-y_j\left(k\right)\right). \tag{5.42}$$

2. Determine the rate of change of the error with respect to the total input signal of each neuron in the current layer (EI):

$$EI_j^{\iota} = \frac{\partial e(k)}{\partial p_j(k)} = EA_j^{\iota} y_j(k)\left(1 - y_j(k)\right). \tag{5.43}$$

3. Determine the rate of change of the error with respect to the weight on the input connection of each neuron in the current layer (EW):

$$EW_{ij}^{\iota} = EI_j^{\iota} y_i(k). \tag{5.44}$$

4. Determine the rate of change of the error with respect to the activity of the neuron in the previous layer (EA):

$$EA_i^{\iota-1} = \frac{\partial e(k)}{\partial y_i(k)} = \sum_{j=1}^{N_{\iota}} EI_j^{\iota} w_{ij}, \tag{5.45}$$

where w_{ij} – the weight of the connection between the j-th neuron in the output layer and the i-th neuron in the input layer.

5. Update the connections between neurons using the gradient rule:

$$w_{ij}\left[k+1\right] = w_{ij}\left[k\right] + \gamma EW_{ij}^{\iota}, \tag{5.46}$$

where γ – the learning rate (iteration step); k – the iteration step number.

Transition to the next layer.

If the specified layer is not the input layer, repeat all procedures starting from **Step 2**.

The training continues until an acceptable error is achieved.

As the activation function in the neural network filter (NNF), a linear function is chosen. This allows, analogously to digital filters, to compute the NNF output in response to the input signal $p(k)$ and its previous values $p(k)$,..., $p(k–L–1)$ using the following formula:

$$y(k) = \sum_{l=0}^{L-1} p(k-l)\hat{w}_l(k) = P^T(k)\widehat{W}(k), \tag{5.47}$$

where $\widehat{W}(k) = \left[\hat{w}_0(k), \hat{w}_1(k), \ldots, \hat{w}_{L-1}(k)\right]^T$ – the column vector of weight coefficient estimates for the generalized characteristic of the radio channel at the k-step; $P(k) = \left[p(k), p(k-1), \ldots, p(k-L-1)\right]^T$ – the column vector of the contents of the delay line of the NNF at the k-step; L – represents the memory of the UAV channel and the number of taps in the delay line at the input of the NNF.

The process of obtaining the values of the UAV channel state assessment is directly related to the training of the neural network filter (NNF).

The training (learning) sequence vector $P(k)$, corresponding to the set of realizations of the random scalar $d(k)$ together form the training dataset for the neural network filter (NNF):

$$T = \left\{P(k), d(k)\right\}_{k=1}^{\Theta}, \tag{5.49}$$

where Θ – the length of the training sequence is denoted.

The estimation error of the received signal $d(k)$ is given by:

$$e(k) = d(k) - y(k) = d(k) - P^{T}(k)\widehat{W}(k). \tag{5.50}$$

Using the learning criterion based on minimizing the mean square error, let's take the cost function for the evaluation as:

$$J\left(\widehat{W}(k)\right) = E_T\left\{e^2(k)\right\}_{k=1}^{\Theta}, \tag{5.51}$$

where E_T – the averaging operator over the entire training dataset T.

To obtain the optimal estimates of the weight coefficients for the neural network filter (NNF), it is possible to use the well-known gradient descent method. For this, it is necessary to compute the instantaneous gradient of the estimation error [9, 19].

An example of obtaining the generalized assessment of the UAV channel state is presented. **Table 5.2** provides the definition of the membership function for the fuzzy terms based on the given indicators, while **Table 5.3** shows an example of forming the generalized (comprehensive) assessment of the UAV channel state.

Table 5.2 Definition of membership functions for fuzzy terms based on given indicators of UAV channel state evaluation

Values of indicators for UAV channel state evaluation	Universal set	Evaluation terms	Membership function
Impulse response	1–10	Channel in normal condition	0.91–1
		Out of range	0.4–0.9
		Channel unfit for use	< 0.3
Frequency response	1–10	Channel in normal condition	0.91–1
		Out of range	0.4–0.9
		Channel unfit for use	< 0.3
Bit error rate	1–10	Channel in normal condition	0.91–1
		Out of range	0.4–0.9
		Channel unfit for use	< 0.3

Table 5.3 Example of forming the generalized (comprehensive) assessment of the UAV channel state

Values of indicators for channel state evaluation	Calculated membership functions	Decision on the channel state
Impulse response	0.080474451	The channel is unfit for operation. There are numerous interferences across the entire bandwidth of the channel. The bit error rate is low and does not allow the channel to be used for transmitting any type of information
Frequency response	0.080474451	
Bit error rate (BER)	0.050553799	

Four types of signals were used in the modeling: phase modulation (FM-4 and FM-8), quadrature amplitude modulation (QAM-16), and signal-code constructions (SCC) with 8 states and 64 points in the constellation.

In the modeling of the method, it is assumed that the method is convergent when, after 15 iterations of the generalized Viterbi algorithm, the error $e(k)$. The error remained below –36 dB after 15 iterations of the generalized Viterbi algorithm [14].

It is possible to conduct a comparative evaluation of the proposed method in comparison with existing ones. The following input data were used in the modeling: MIMO 8×8 configuration, modulation type – phase modulation, signal ensemble dimensionality $M = 256$; and error-correcting code type – convolutional codes with a rate $R = 0.9$.

The comparison was made using the Zero Forcing (ZF) method, the algorithm optimized according to the Minimum Mean Square Error (MMSE) criterion, and the Maximum Likelihood (ML) method.

The comparative analysis of the proposed method with known methods shows that the proposed method allows for an average 30 % increase in the speed of channel state estimation in MIMO systems, thus improving the interference immunity of UAV channels.

Fig. 5.4 shows the focused neural network filter built based on a linear adaptive summator, with its operating procedure for evaluating the state indicators of UAV channels. **Fig. 5.5** shows one of the design implementations of the focused neural network filter for UAV channels.

The method proposed in the work is advisable to use when developing software for modules (blocks) that evaluate prospective radio communication systems, based on the open architecture interfaces of version SCA 2.2.

5.2 DEVELOPMENT OF THE METHODOLOGY FOR IDENTIFYING THE STATE OF CONTROL AND DATA TRANSMISSION CHANNELS OF UNMANNED AERIAL VEHICLES (UAVS)

Let the output of the UAV channels be represented by measurements that form a sample of volume s, and $\{y_i, t_i\}, i = \overline{1, s}$, where $y_i \in R$ – the measurements of the UAV channels' output at a given moment in time $t_i \in [0, +\infty)$, $u = u(t)$ – the known data controls the input of the UAV channels.

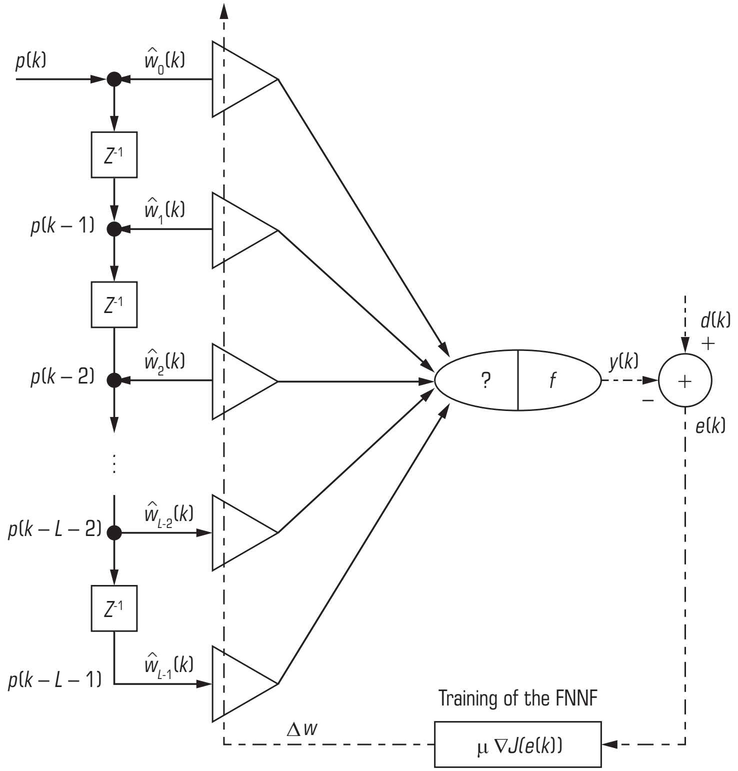

Fig. 5.4 Focused neural network filter built on the basis of a linear adaptive summator

It is also known that the system is linear and is described by a linear differential equation of the form, and the initial condition of the equation is known:

$$a_k \cdot x^{(k)} + a_{k-1} \cdot x^{(k-1)} + \ldots + a_0 \cdot x = b \cdot u(t), \qquad (5.52)$$
$$x(0) = x_0.$$

It is necessary to determine the system parameters and the order n of the differential equation from the sample data, which let's consider as bounded, i.e., $n - 1 \le M$, $M \in N$. It is assumed that an additively distributed symmetric noise $\xi : M(\xi) = 0$, $D(\xi) \langle \propto$, and $y_i = x(t_i) + \xi_i$.

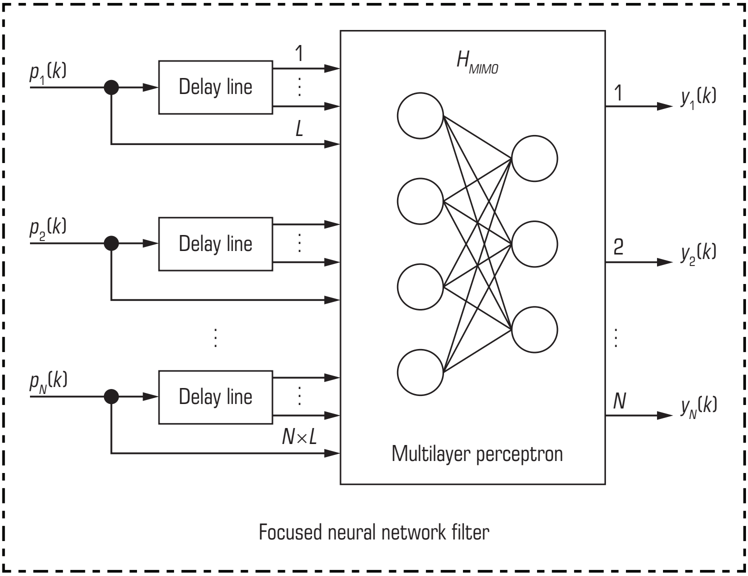

Fig. 5.5 Implementation of the neural network filter (NNF) for the MIMO transmission scheme

Thus, when the order of the system is unknown, we are solving the structural-parametric identification problem. The task will be partially parameterized since the maximum derivative degree that enters the equation is determined in advance, limiting the dimensionality of the search space. This formulation was first discussed in [20] and further developed in [21].

It is possible to assume that for a system of any order, its coefficient at the highest degree equals 1, as follows:

$$x^{(k)} + \frac{a_{k-1}}{a_k} \cdot x^{(k-1)} + \ldots + \frac{a_0}{a_k} \cdot x = \frac{b}{a_k} \cdot u(t), \tag{5.53}$$

or

$$x^{(k)} + \tilde{a}_k \cdot x^{(k-1)} + \ldots + \tilde{a}_1 \cdot x = \tilde{b} \cdot u(t). \tag{5.54}$$

Then, the solution to the identification problem will be sought as a differential equation of order $m \le M$, $M \in N$, such that the solution to the Cauchy problem with given initial conditions:

$$x^{(k)} + \hat{a}_k \cdot x^{(k-1)} + \ldots + \hat{a}_1 \cdot x = \hat{b} \cdot u(t),$$
$$\hat{x}(0) = x_0, \tag{5.55}$$

with parameters $\hat{a} = \begin{pmatrix} 0 \ldots 0 & \hat{a}_m \ldots \hat{a}_1 & \hat{a}_0 \end{pmatrix}^T \in R^n$, i.e., $n = M + 1$, which deliver the extremum of the chosen function.

$$I_1(a) = \sum_{i=1}^{N} \left| y_i - \hat{x}(t_i) \right| \Big|_{\hat{a}=a} \to \min_{a \in R^n},$$
$$I_2(a) = \max_i \left| y_i - \hat{x}(t_i) \right| \Big|_{\hat{a}=a} \to \min_{a \in R^n}. \tag{5.56}$$

To estimate the process model in the form of (5.53) as a solution to one of the extremum search problems: (5.54) or (5.55), it is necessary to have information about the initial position of the system so that the Cauchy problem can be solved.

In general, the vector of the initial position of the system, if it is not known initially, can be numerically estimated, which, of course, is not always possible and depends on the properties of the sample. Another option for determining the initial position of the system is to include the vector in the optimization task. Typically, for many tasks, the observation process starts from a steady state, so all coordinates of the initial position for UAV channels are set to 0, except for the output position of the UAV channels.

It is possible to assume that it is necessary to build a mathematical model of the dynamic process, which it is possible to represent in matrix form:

$$\tilde{x}' = \hat{A} \cdot \tilde{x}(t) + \hat{B} \cdot u(t), x(0) = x_0, \tag{5.57}$$

where $\hat{A} = \left(\hat{a}_{ij}\right)_{i=1,j=1}^{n,n}$ – the matrix of the system of linear differential equations is represented as: $\hat{B} = \left(\hat{b}_{ij}\right)_{i=1,j=1}^{n,m}$ – the matrix of the right-hand sides, control coefficients; $\tilde{x}(t) \in R^n$ – state model of the UAV channels; $u(t) \in R^m$ – control actions, represented as a vector function.

Given that several different outputs of the UAV channels are observed, which may differ in the amplitude of the response, it is necessary to normalize each individual criterion. To do this, it is possible to define the diameter of the measurement set for each observed output by including the initial position of the output in this set. Then, the criterion takes the following form:

$$I(a) = \sum_{j=1}^{N_0} \frac{\sum_{i=1}^{s_j} \left| y_i^j - \hat{x}^j\left(t_i^j\right) \right|}{\sup\left(|a-b| : a,b \in Y^j \bigcup x_0^j \right)} \Bigg|_{\hat{A}=A,\hat{B}=B} \to \min_{A,B}, \tag{5.58}$$

where N_0 – the number of UAV channel outputs; $s_j, j=\overline{1,N_0}$ – the sample size for each UAV channel output; $y_i^j,\ i=\overline{1,\bar{s}_j},\ j=\overline{1,N_0}$ – the measurements of the outputs forming the samples; $t_i^j,\ i=\overline{1,\bar{s}_j},\ j=\overline{1,N_0}$ – the measurement times for each j-th output; $\sup\left(\left|a-b\right|: a,b\in Y^j \bigcup x_0^j\right)$ – the diameter of the measurement set for each output; $\hat{z}^j(t)\Big|\hat{A}=A,\hat{B}=B$ – j-th output of the model with matrices A, B.

Criterion (5.58) is analogous to the criterion for a system with one input and one output, as in (5.56). Thus, the task of identifying the UAV channels has been reduced to searching for the extremum in the space of vectors with real coordinates. At the same time, the peculiarities of representing the structure of UAV channels lead to complex behavior of the objective function near certain points of the space, where the first coordinates of the vector approach 0.

The methodology for identifying the state of the control and data transmission channels of UAVs consists of the following interconnected procedures:

1. Input of initial data about the state of the UAV channels.

2. Initialization of the initial model based on expressions (5.52)–(5.58).

3. Input of correction coefficients for noise and prior uncertainty about the state of the UAV channels using the expressions [22].

Given the lack of prior information about the coefficients and the order of the differential equation, using binary representation of the optimization variables becomes difficult and inefficient in terms of finding a solution.

According to the accepted transition from the vector, i.e., the individual, to the differential equation, the vector, given the specifics of the chosen representation of the solution, contains information about the order, structure, and coefficients of the differential equation, which must be taken into account for improving the algorithm's performance.

4. Determination of the order of the differential equation.

Let $\hat{a}$ – be the vector containing the solution to the problem, then $i_{order}\in N, i_{order}\le M, i_{order}:$ $\hat{a}_{i_{order}}\ne 0, \hat{a}_i=0, i\rangle i_{order}$. If $\hat{a}_M\ne 0\to i_{order}=M$. Then the order of the equation will be determined by the index i_{order}. Considering the proposed approach to determining the order of the differential equation, it is important that the algorithm for solving the problem has the ability to maintain certain coordinates equal to 0.

5. Rounding of vector coordinates.

One of the special modifications of the algorithm was the introduction of the rounding operation for vector coordinates:

$$op_j^i = round\left(op_j^i\right),\ j=\overline{1,n},\ i=\overline{1,N_1}, \tag{5.59}$$

where $round(\cdot): R\to Z$ – a function that rounds a number to its nearest integer. This operator, which influences the objective parameters of the algorithm, solves the task of converting the vector coordinates into integers. Since, for representing the system's structure, it is important that in some cases a certain number of solution coordinates consecutively approach zero, and the stochastic search algorithm disturbs variables due to the nature of the mutation operation, an operator is needed that would maintain the found order. The rounding operator is applied immediately after the mutation operator, and after rounding, a local improvement of the obtained population occurs.

6. Mutation of individuals in the population.

To enhance the effectiveness of the solution search regarding the state of a heterogeneous dynamic object, the mutation operator was modified in such a way that the mutation probability for each pair of objectives – strategic parameters is $p_m = \frac{1}{q}$. Thus, random disturbances do not lead to a significant spread of individuals in the next population around a certain found solution, which is not eliminated during the subsequent local improvement of the alternative after rounding.

7. Generation of the initial population.

Since random selection of coefficients for the initial population will not lead to the appearance of diverse solutions that correspond to equations of different orders, considering such a representation of solutions, the population will be generated as follows:

1. For each individual, with probability $\frac{1}{M}$ the order of the differential equation is selected.
2. For the chosen order i_{order} each non-zero coordinate is solved uniformly over the interval [–5, 5].
3. All strategic parameters of the individual are randomly selected uniformly over the interval [0, 1].

The proposed scheme was selected as the best through the trial-and-error process of different initial solution generation variants.

It is necessary to account for some specifics of the local improvement of alternatives using the proposed random coordinate descent algorithm. The rounding operator (5.59) applied to each coordinate results in the loss of solution accuracy due to the truncation of the mantissa. To compensate for the loss of precision and improve the overall efficiency of the algorithm, it is necessary for the coordinate descent to take such a number of steps that, with the chosen step length, the rounded coefficient could be refined in such a way that the value returned would precede the integer.

End.

To evaluate the effectiveness of the proposed methodology, 100 systems were randomly generated: 10 systems for each order of the differential equation, from the first to the tenth. The parameters of each system were randomly selected as follows: $\hat{a}_k^i \sim U(-5,5)$, $\hat{b}_k \sim U(-5,5)$, $i = \overline{1,10}$, $k = \overline{1,i}$. The operating time of the system was chosen to be 5.

The control function for all analyzed tasks was chosen to be a unit function, i.e., $u(t) = 1$. The sampled data are selected from the numerical solution of the differential equation. $\{x_i, t_i\}$, $i = \overline{1, T / h_{ode}}$ – the numerical solution of the system. Then, for the given sample size $s\langle T / h_{ode}$, $s = 100$ select s different points are randomly selected from the numerical solution of the differential equation.

To evaluate the effectiveness of the optimization algorithm's parameters, the identification of UAV channels without disturbances in the measurement channels was considered, so that this factor would not introduce additional complexity into the task and the obtained solutions could be assessed. For this reason, the sample size was chosen to be sufficiently large to assess its representativeness, i.e., to ensure that the sampled data encompassed all the features of the transient process.

For each individual system, 20 runs of the algorithm were performed with specific settings. All initial conditions for these tasks were set to zero. The population size was chosen to be 50, and the number of populations was set to 50. The parameters of the local descent were $N_1 = 50$, $N_2 = 50$, and $N_3 = 1$ if $h_l = 0.05$.

The study of the effectiveness of the proposed methodology showed that the average fitness increases as the order of the real object approaches the established parameter-limit of the maximum model order. This suggests that the algorithms should work in such a way as to retain the ability to reduce the system order while maintaining the equality of the first coordinates at 0. Therefore, modifying the mutation or introducing the rounding operator leads to significant improvement. It is also important to note that the increase in fitness is related to the fact that, over the observation period, a higher-order system behaves in such a way that it is easier to build its model than, under similar conditions, a lower-order model.

The transient processes of different systems may coincide over a certain interval, so only increasing the observation interval for the system output and increasing the frequency of measurements can improve the efficiency of finding a solution. On the other hand, this may be due to the presence of a large number of local optima and a strong attraction zone.

CONCLUSIONS

1. A method for the intellectual evaluation of the state of UAV channels is proposed, based on the use of fuzzy sets and artificial neural networks.

Despite its simplicity, the method allows obtaining quite accurate solutions. Key features of the proposed method include:

– state of the UAV channels is evaluated in parallel across several indicators of their state (impulse response, frequency response, and bit error probability);

– evaluation of several channel state characteristics of UAVs is continuously performed in real-time;

– evaluation of several characteristics of the UAV channel's state is continuously carried out in both the downlink and uplink channels;

– the state of the UAV channel is evaluated for each indicator on a separate layer of the neural network using the construction of the membership function;

– after evaluating each channel characteristic on a separate neural network layer, a generalized assessment of the channel's state is formed at the output.

The evaluation obtained using the proposed intellectual evaluation method matches the results obtained using the optimal algorithm based on the minimum mean square error criterion. Additionally, these results are computed 30 % faster, which reduces the adaptation time of the radio communication device. A key factor determining the effectiveness of the proposed comprehensive method is the degree of training of the neural network to the signal environment.

2. To reduce the neural network training time and increase the efficiency of the proposed method, it is advisable to preload the knowledge bases of the signal environment. This will minimize the network training time and simplify the adaptation process of the radio communication device by an average of 15 %.

3. The proposed method for the intellectual evaluation of the UAV channel's state can be implemented in programmable architecture-based communication devices. To achieve this, it is necessary to adapt the signal processor through additional software for the specific transceiver device. This software should be developed on the SCA 2.2 platform.

4. The study presents the development of a methodology for identifying the state of UAV control and data transmission channels.

The novelty of the proposed methodology lies in:

– accounting for the corrective coefficient based on the degree of uncertainty regarding the state of UAV channels in the calculations;

– adding a corrective coefficient for the noise in the data resulting from the distortion of information about the UAV channel's state;

– reducing computational costs when evaluating the state of UAV channels;

– ability to perform calculations with input data that differ in nature and units of measurement.

This methodology is recommended for implementation in specialized software used for analyzing the state of complex technical systems and making management decisions.

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 deci-

sion 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. Engineer-

ing 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. 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

DOI: 10.15587/978-617-8360-13-9.CH6

Svitlana Kashkevich, Oleksandr Yaryzhko,
Olena Nosyryeva, Anzhela Popova, Ihor Pimonov, Dmitry Leinyk

CHAPTER 6

INTELLIGENT METHODS FOR EVALUATING THE STATE OF HIERARCHICAL SYSTEMS

ABSTRACT

This section of the study proposes intelligent methods for assessing the state of hierarchical systems.

During the research, the authors:

– conducted an analysis of knowledge representation models, substantiating the advantages of using production knowledge representation in expert systems. The study outlines key concepts of fuzzy expert systems and formulates a formal task for accelerating decision-making in the rule base of a fuzzy expert system;

– developed a methodology for assessment and prediction using fuzzy cognitive maps.

The novelty of the proposed methodology lies in:

– considering a corrective coefficient for the degree of uncertainty regarding the state of the object;

– adding a corrective coefficient for data noise resulting from distortions in object state information;

– reducing computational costs when evaluating object states;

– creating a multilevel and interconnected description of hierarchical objects;

– adjusting the object description due to changes in its current state using a genetic algorithm;

– enabling calculations with input data of different natures and measurement units.

The research includes the development of a visualization method for hierarchical system states. The novelty of this method lies in:

– creating a visual, multilevel, and interconnected description of the hierarchical system;

– enhancing decision-making efficiency in assessing the hierarchical system's state;

– addressing the issue of global and local extrema when evaluating the state of hierarchical systems;

– combining graphical and numerical representations of the monitored parameters of the hierarchical system's state;

– avoiding loop formation during real-time visualization of hierarchical system states.

The study also develops a method for evaluating complex hierarchical systems based on an improved particle swarm optimization (PSO). This evaluation method combines PSO with coordinate averaging and its modification by employing multiple particle swarms and integrating the Hooke-Jeeves procedure and appropriate corrective coefficients.

The novelty of this method lies in:
– creating a multilevel and interconnected description of real-time complex hierarchical systems;
– enhancing decision-making efficiency for real-time hierarchical systems assessment;
– resolving global and local extrema issues during real-time hierarchical system state evaluation;
– enabling directed searches by multiple swarm particles in a specific direction;
– considering the degree of uncertainty;
– allowing for repeated analysis of complex real-time hierarchical systems.

KEYWORDS

Artificial intelligence, heterogeneous data processing, hierarchical systems, reliability, efficiency.

Optimization is a complex process of determining a set of solutions for various functions. Many computational tasks today belong to the domain of optimization problems [1–3]. When solving optimization problems, decision variables are defined to ensure that a hierarchical system operates at its optimal point (or mode) based on a specific optimization criterion.

Optimization problems in hierarchical systems are often discontinuous, non-differentiable, and multimodal. Therefore, classical gradient-based deterministic algorithms [4–6] are not suitable for addressing the optimization tasks of organizational and technical systems.

To overcome the limitations of classical optimization algorithms in solving hierarchical system optimization problems, a significant number of stochastic optimization algorithms, known as metaheuristic algorithms, have been developed [7–11].

One type of stochastic optimization algorithm for hierarchical systems is swarm intelligence algorithms (swarm-based algorithms). These algorithms are based on the movement of a swarm and simulate its interaction with the environment to improve knowledge about the surroundings, such as discovering new food sources. The most well-known swarm algorithms include Particle Swarm Optimization (PSO), Artificial Bee Colony Optimization, Ant Colony Optimization, Wolf Pack Optimization, and Sparrow Search Algorithm [12–18].

Unfortunately, most of the aforementioned basic metaheuristic algorithms fail to balance exploration and exploitation, leading to unsatisfactory performance in solving real-world complex optimization problems [19–38].

This has motivated the development of various strategies to enhance the convergence speed and accuracy of basic metaheuristic algorithms [39–69]. One approach to improving the efficiency of decision-making with metaheuristic algorithms is their hybridization, where fundamental procedures of one algorithm are incorporated into another [70–86].

Given the above, the development of intelligent methods for assessing the state of hierarchical systems using artificial intelligence is a pressing scientific challenge. Such methods would significantly improve the efficiency of decision-making processes.

6.1 ANALYSIS OF ADVANTAGES AND DISADVANTAGES OF KNOWLEDGE REPRESENTATION METHODS IN INTELLIGENT DECISION SUPPORT SYSTEMS

This section analyzes the advantages and disadvantages of knowledge representation methods in intelligent decision support systems.

The logical model is used for knowledge representation in first-order predicate logic systems and for deriving conclusions through syllogisms.

The main advantage of using predicate logic for knowledge representation lies in its powerful inference mechanism, which possesses well-understood mathematical properties and can be directly programmed. With these programs, new knowledge can be derived from previously known facts.

Distinctive features of logical models include the unambiguity of theoretical justification and the possibility of implementing a system of formally accurate definitions and conclusions.

The main idea behind constructing logical knowledge models is as follows: all information necessary for solving applied tasks is considered as a set of facts and statements presented as formulas in a certain logic. Knowledge is represented as a set of such formulas, and obtaining new knowledge is reduced to implementing logical inference procedures.

Main advantages of logical knowledge models:

1. The foundation of logical models is the classical apparatus of mathematical logic, which methods are well-studied and formally justified.

2. Efficient inference procedures are available, including those implemented using logical programming languages.

3. Knowledge bases can store only sets of axioms, while other knowledge is derived from them using inference rules.

Frame model. The frame model, or knowledge representation model, is a systematic representation of human memory and cognition.

Frames take the form of structured components of situations, called slots. A slot may refer to another frame, thereby establishing a connection between two frames. General relationships, such as communication links, can also be established. Each frame is associated with diverse information (including procedures), such as expected procedures for a situation, ways of obtaining information about slots, default values, and inference rules.

Advantages of the frame model – representation is largely based on including assumptions and expectations. This is achieved by assigning default values to slots in standard situations. During the search for solutions, these values can be replaced with more reliable ones. Some variables are designed in such a way that the system must ask the user about their values.

Frame models ensure structural organization and connectivity. This is achieved through inheritance and nesting properties of frames, meaning that a slot may represent a system of lower-level slot names, and slots may be used as calls for executing procedures. Slot values may be refined during the processing of knowledge represented in this model. Some variables may be defined as embedded procedures. As values are assigned to the variables, other procedures are triggered. This type of representation combines declarative and procedural knowledge.

For many subject areas, frame models are the primary method of formalizing knowledge.

Semantic networks. A semantic network is a directed graph structure where nodes represent concepts (objects, processes, situations) and edges correspond to relationships such as "is a", "belongs to", "is caused by", "is part of", "is composed of", "is like", and similar connections between pairs of concepts. Reasoning procedures used in semantic networks include network augmentation, property inheritance, and pattern matching. Distinctive features of semantic networks include class-element relationships (part-whole, class-subclass, element-set), property-value relationships (having a property, having a value), and examples of class elements (element above, element below, earlier, later).

The advantages of semantic networks include the clarity of knowledge representation, which makes it convenient to depict causal relationships between elements (subsystems) and even the structure of complex systems. The disadvantage of such networks is the complexity of reasoning and searching for subgraphs that match a specific query.

Production model. Rule-based systems are the most widely used type of expert systems today. In rule-based systems, knowledge is not represented in a declarative, static manner (as a series of true statements) but rather in the form of numerous rules that specify conclusions to be drawn or not drawn in various situations. A rule-based system consists of "IF-THEN" rules, facts, and an interpreter that manages which rule should be invoked based on the facts in the working memory.

A classical expert system embodies informal knowledge obtained from an expert.

This process of creating an expert system is called knowledge engineering and is performed by a knowledge engineer.

Rule-based systems are categorized into two main types: forward-chaining systems and backward-chaining systems. Forward-chaining systems start with known initial facts and proceed by using rules to infer new conclusions or perform specific actions. Backward-chaining systems start with a hypothesis or goal that the user aims to prove and work backward by finding rules that can confirm the truth of the hypothesis. To break down a large task into smaller, manageable fragments, new subgoals are created. Forward-chaining systems are primarily data-driven, while backward-chaining systems are goal-driven.

The working memory can contain facts about the current state of the object. A rule whose conditions are all satisfied is called activated or fired. The working list of rules may contain several activated rules. In such cases, the inference engine must choose one of the rules to execute.

The THEN part of a rule contains a list of actions to be performed once the rule is executed.

The inference engine operates in a cycle of checking and executing rules. During each cycle, multiple rules may be activated and added to the working list of rules.

Conflicts arise in the rule list when different activated rules have the same priority, requiring the inference engine to decide which rule to execute. Once all rules are executed, control is returned to the command interpreter so that the user can issue additional instructions to the expert system's command interpreter.

A notable feature of an expert system is its explanation mechanism, which allows the user to ask questions about how the system reached a specific conclusion and why certain information is needed. Rule-based systems can easily answer questions about how a conclusion was derived because the rule activation chronology and the content of the working memory can be stored in a stack.

The widespread use of rule-based systems is due to the following reasons: modular organization simplifies knowledge representation and allows for incremental development of the expert system. Explanation capabilities enable the easy creation of explanation tools through rules, as the antecedents of rules specify what is required for rule activation. The explanation tool allows tracking which rules were executed, enabling the reconstruction of the reasoning process that led to a specific conclusion.

Analogy with human cognitive processes: according to findings by Newell and Simon, rules naturally align with the way humans solve problems. When extracting knowledge from experts, it is easier to explain the structure of knowledge representation to experts because rules provide a simple and intuitive framework.

Fuzzy expert systems. The foundation of the functioning of expert systems lies in the knowledge model [9]. It contains a set of principles that describe the state and behavior of the object under study. The most widely used knowledge model of expert systems is the production model, as it is quite simple to process and understandable to the end user.

However, recently fuzzy expert systems have become widespread [11]. This type of expert system is based on a set of rules in which linguistic variables and fuzzy relationships are used to describe the state and behavior of the object under study [12]. Rules presented in this form are closest to natural language, so there is no need to use a separate knowledge engineer to create and edit the rules. They can be edited by the expert themselves almost without special training. Also, the results of such systems are presented in a limited natural language, which increases their degree of adaptation to the end user. Let's consider the organization of fuzzy expert systems in more detail.

A fuzzy expert system uses knowledge representation in the form of fuzzy productions and linguistic variables [13]. Each linguistic variable is defined using its term set, which consists of fuzzy variables [14].

Fuzzy variable. The concept of a fuzzy variable is used to describe objects and phenomena through fuzzy sets, where the membership of a specific element is defined by a membership function $\mu z(u)$, this function determines the degree to which the value u belongs to the set z [15, 17, 18]. Every fuzzy variable is characterized by a triplet $<z, U, Z>$, where z – the name of the variable, U – the universal set, Z – the fuzzy subset of U, which represents the fuzzy constraint on the value $u \in U$, as defined by z [19].

Linguistic variable. The operation of fuzzy expert systems is based on the concept of a linguistic variable [15]. Each linguistic variable has a set of values, which are fuzzy variables forming its term set [6, 14].

Linguistic Variable L is characterized by the following set of properties:

$$L=(X, T(X), U, G, M), \tag{6.1}$$

where X – the name of the variable; $T(X)$ – the term set of variable X, which is a set of linguistic values for X, each value is a fuzzy variable (x') with values from the universal set U associated with a base variable u; G – the syntactic rule that generates the names of the values of variable X; M – the semantic rule that assigns each fuzzy variable x' its meaning $M(x')$, which is a fuzzy subset $M(x')$ of the universal set U.

Fuzzy rule base of an expert system. The behavior of the studied system is described in a limited natural language using linguistic variables [17, 18]. The input and output parameters of the system are treated as linguistic variables, and the process is described by a set of rules [9, 10]. The formal model of the rule base for the developed expert system is represented as follows [11, 12]:

$$\begin{aligned}
&R_1: A_{1,1} \text{ and/or } A_{1,2} \text{ and/or } \ldots \text{ and/or } A_{1,m1} \to B_{1,1} \text{ and/or } \ldots \text{ and/or } B_{1,k1},\\
&R_2: A_{2,1} \text{ and/or } A_{2,2} \text{ and/or } \ldots \text{ and/or } A_{2,m2} \to B_{2,1} \text{ and/or } \ldots \text{ and/or } B_{2,k2},\\
&R_n: A_{n,1} \text{ and/or } A_{n,2} \text{ and/or } \ldots \text{ and/or } A_{n,mn} \to B_{n,1} \text{ and/or } \ldots \text{ and/or } B_{n,kn},
\end{aligned} \tag{6.2}$$

where $A_{i,j}$, $i=1,2,\ldots,n$, $j=1,2,\ldots,m_i$ – fuzzy statements defined on the values of input linguistic variables; $B_{i,q}$, $i=1,2,\ldots,n$, $q=1,2,\ldots,k_i$ – fuzzy statements defined on the values of output linguistic variables.

In general, fuzzy reasoning is carried out in four stages [13–15]:

1. Fuzzification stage: conversion of precise input data into fuzzy values of linguistic variables using membership functions.

2. Fuzzy inference stage: based on the set of rules in the fuzzy knowledge base, the truth values for the conditions of each rule are calculated using T-norm, T-conorm, and negation operations.

3. Composition stage: values of output linguistic variables are formed for each rule that has been triggered.

4. Defuzzification stage: conversion of fuzzy values of output linguistic variables into precise values.

Fuzzy logical inference. Let examine the stages of fuzzy decision-making in detail [19]:

1. *Fuzzification Stage*. Using the membership functions of all terms for input linguistic variables and based on precise input values from the universes of input linguistic variables, the degrees of confidence are determined. These represent the likelihood that the output linguistic variable assumes specific values [11, 12].

2. *Fuzzy Inference Stage*. From the set of rules (fuzzy knowledge base), the truth value for each rule is calculated based on specific fuzzy operations corresponding to the conjunction or disjunction of terms in the left-hand side of the rules. Typically, the maximum or minimum of the confidence degrees of terms, determined during fuzzification, is applied to the conclusion of each rule. Using one of the methods for constructing fuzzy implication, a fuzzy variable is generated that

corresponds to the computed confidence degree in the left-hand side of the rule and the fuzzy set in the right-hand side [13, 14].

3. *Composition (Aggregation or Accumulation) Stage*. All fuzzy sets assigned to each term of each output linguistic variable are combined into a single fuzzy set, representing the value of each output linguistic variable derived. This is typically achieved using maximum or summation functions [15, 16].

4. *Defuzzification Stage.* Defuzzification is applied when it is necessary to transform a fuzzy set of derived linguistic variable values into precise ones. There are numerous methods for this transition, with the most commonly used being the full interpretation method and the maximum interpretation method.

Full Interpretation Method: the precise value of the output variable is computed as the "center of gravity" of the membership function for the fuzzy value.

Maximum Interpretation Method: the precise value of the output variable is determined as the maximum value of the membership function [17–19].

The highest computational costs are incurred during the fuzzy inference stage. To address this, the study examines a proposed method for accelerating the decision-making process at this stage [10, 11].

Task of accelerating decision-making in fuzzy expert systems. To formulate the task of accelerating decision-making, the concept of a single iteration of fuzzy inference is introduced [12, 13]. It is proposed to represent it as a function F, which transforms a set of conditions into a set of consequences, as follows:

$$F:\{A_{1,1},A_{1,2},\ldots,A_{1,m1},A_{i,1},A_{i,2},\ldots,A_{i,mi},A_{n,1},A_{n,2},\ldots\ ,A_{n,mn}\}\rightarrow$$
$$\rightarrow\left\{B_{1,1},B_{1,2},\ldots\ ,B_{1,k1},B_{i,1},B_{i,2},\ldots,B_{i,ki},B_{n,1},B_{n,2},\ldots\ ,B_{n,kn}\right\}. \tag{6.3}$$

The task of accelerating decision-making is to minimize the computations performed during the processing of the condition matrix A for fuzzy rules, this involves constructing a reduced set of fuzzy conditions A*, that $|A^*|<|A|$, ensuring that the result remains consistent. Specifically, if $F(A)\rightarrow B$, then $F(A^*)\rightarrow B$.

Acceleration of decision-making can be achieved in two ways [14–16]:

1. Exclusion of certain rules from processing.

Suppose certain rules are excluded $i1,\ i2,\ldots,\ is$. In this case:

$$|A^*|=|A|-\sum_{t=1}^{S}(m_{it}). \tag{6.4}$$

2. Identifying identical conditions and eliminating their redundant computation: assume that p matches of conditions of the form $A_{i,j}=A_{v,w}$, where $i=1,2,\ldots,n,\ j=1,2,\ldots,m_n,\ v=1,2,\ldots,n,$ $w=1,2,\ldots,m_n$. Where:

$$|A^*|=|A|-p. \tag{6.5}$$

The input data for the proposed method are the rules from the fuzzy rule base.

The following has been established:

– the methods (models, approaches) for knowledge representation in intelligent decision support systems presented in the study, in their canonical form, are not feasible for use due to a number of objective reasons outlined in Section 3 of the study;

– it is necessary to develop new (or improve existing) methods of knowledge representation in intelligent decision support systems that combine the advantages of these approaches without their disadvantages.

Future research should focus on further refinement of these approaches to reduce the number of drawbacks and limitations in their application.

6.2 DEVELOPMENT OF AN EVALUATION AND PREDICTION METHODOLOGY USING FUZZY COGNITIVE MAPS

The methodology for evaluation and prediction using fuzzy cognitive maps includes the following interrelated procedures:

1. *Input of Initial Data on the Object's State:* collecting the primary data about the object's condition.

2. *Initialization of the Object's Initial State Model:* setting up the base model to reflect the initial state of the object.

3. *Introduction of Corrective Coefficients for Noise and A Priori Uncertainty:* adjusting for noise and uncertainty using specific expressions [2].

Due to the absence of prior information about coefficients and the order of the differential equation, binary representation of optimization variables becomes challenging and inefficient for finding solutions. However, when information about data noise and uncertainty levels regarding the object's state is available, it becomes possible to improve the accuracy of constructing fuzzy cognitive maps.

4. *Construction of a fuzzy cognitive map of the object's state.*

The transition from a vector (individual) to a differential equation must consider the order, structure, and coefficients of the equation, as these influence the optimization algorithm's performance [11–15].

To construct a cognitive map reflecting the dynamic properties of the situation, scales for factor values and their increments must be defined.

The scale of a factor is determined by structuring the set of its linguistic values. Absolute values of factors, rather than subjective assessments like "large", "medium", or "small" are used to define linguistic values. An objective reference value (benchmark) for the factor is established as the standard for its value. This reference simplifies the expert's work in assessing the impact strength of factors and reduces errors.

The prediction involves the max-triangulation composition of the weight matrix and the vector of initial factor increments.

This algorithm is designed for positive-definite matrices. However, in this case, elements of the adjacency matrix and increment vectors may take both negative and positive values.

The following rule is used for transforming the adjacency matrix: $W = \left|w_{ij}sl\right| n \times n$ with positive and negative elements into a positively defined double matrix $W = \left|w'_{ij}sl\right| 2n \times 2n$:

$$\text{if } \left(w_{ij}sl\right) > 0, \text{ then } w'_{i(2j-1)}s_{(2l-1)} = w_{ij}sl,\ w'_i\left(2_j\right)s\left(2l\right) = w_{ij}sl; \tag{6.6}$$

$$\text{if } \left(w_{ij}sl\right) < 0, \text{ then } w'_{i(2j)}s_{(2l)} = -w_{ij}sl,\ w'_i\left(2_j\right)s\left(2l-1\right) = -w_{ij}sl. \tag{6.7}$$

Initial increment vector $P(t)$ and the vector of predicted feature values $P(t+1)$ in this case must have a dimensionality of $2n$. The rule for obtaining the initial increment vector $P'(t)$ of dimensionality $2n$ from the initial vector $P(t)$ of dimensionality n is as follows:

$$\text{if } p_{ij}\left(t\right) > 0, \text{ then } p'_i\left(2_j - 1\right)\left(t\right) = p_{ij}\left(t\right),\ p'_i\left(2_j\right)\left(t\right) = 0; \tag{6.8}$$

$$\text{if } p_{ij}\left(t\right) < 0, \text{ then } p'_i\left(2_j\right)\left(t\right) = p_{ij}\left(t\right),\ p'_i\left(2_j - 1\right)\left(t\right) = 0. \tag{6.9}$$

In the vector $P'\left(t\right) = \left(p_{11}-, p_{11}+, \ldots, p_{nm}-, p_{nm}+\right)$ the value of the feature f_{ij} characterizes two elements: the element with index 2_j characterizes the positive $p_{ij}+$, and the element with index $2_j - 1$ – negative p_{ij} – the increment of the feature f_{ij}. Then the double increment vector $P'\left(t+1\right)$ for the positively defined matrix W' is determined using the following equation:

$$P'\left(t+1\right) = P'\left(t\right) \circ W', \tag{6.10}$$

where to calculate the element of the vector $P'\left(t+1\right)$ the following rule is used:

$$p'_{ij}\left(t+1\right) = \max_{sl}\left(p'sl\left(t\right) \cdot w'_{ij}sl\right). \tag{6.11}$$

The elements of the increment vectors of feature values, obtained at successive moments in time $P'\left(t+1\right), \ldots, P'\left(t+n\right)$, after transposition, are represented as a block matrix:

$$P_t = \left|P'\left(t+1\right)T, \ldots, P'\left(t+n\right)T\right|. \tag{6.12}$$

The rows of this matrix represent the increments of a single feature over successive moments in time, while the columns represent the increments of all features at the moment corresponding to the selected column. The matrix P_t is called the increment matrix and is used in the operation of algorithms for explaining the predictions of situation development [16–18].

5. *Forecasting the dynamics of an object's state.*

A set of situation factors $F=\{f_j\}$, $j=1,\ldots,m$; $Z_j=\{z_{jk}\}$ – given as an ordered set of linguistic values for the *i*-th factor, *k* – denotes the index of the linguistic value, and the scales for all factors are defined X_i.

The cognitive map is determined by experts (*F*, *W*), where *F* – a set of vertices representing the situation factors, $W=|w_{ij}|$ – an adjacency matrix, the initial state of the situation is represented as a vector containing the values of all situation factors $X(0)=(x_{10},\ldots,x_{m0})$. The initial increment vector of the situation factors is $P(t)=(p_1,\ldots,p_m)$.

It is necessary to find the state vectors of the situation $X(t),X(t+1),\ldots,X(t+n)$ and the increment vectors of the situation's state $P(t),P(t+1),\ldots,P(t+n)$ at successive discrete moments in time t, $t+1$, …, $t+n$, where t – the step number (iteration) of the modeling process.

The forecast of situation development is determined using the matrix equation:

$$P(t+1)=P(t)^{\circ}W,$$

where (°) – rule *max-product*: $p_i(t+1)=\max\limits_j\left(p_j(t)w_{ij}\right)$.

An element of the forecast vector for situation development $p_i(t+1)\in P(t+1)$ is represented as a pair: $\langle p_i(t+1),c_i(t+1)\rangle$, where $p_i(t+1)$ is the increment value of the factor; $c_i(t+1)$ is the consonance of the factor value. Cognitive consonance characterizes the subject's confidence in the modeling results. When $c_i(t)\approx 1$ the subject's confidence in the increment of factor $p_i(t)$ is maximal, and when $c_i(t)\approx 0$ is minimal.

The state of the situation at successive time points is represented as a pair: $\langle X(t+1),C(t+1)\rangle$, where $X(t+1)=X(t)+P(t+1)$ is the state vector of the situation (with element $x_i(t+1)=$ $=x_i(t)+p_i(t+1)$), the cognitive consonance of the factor values $c_i(t+1)\in C(t+1)$.

A plausible forecast of situation development in this case is defined as a pair $\langle X(m),C(m)\rangle$, where $X(m)=(x_1(m),\ldots,x_m(m))$ is the vector of situation factor values at time $t=m$; $C(m)=$ $=(c1(m),\ldots,cm(m))$ is the vector of consonance values for the situation factors at time $t=m$.

6. *Training a fuzzy cognitive map using a genetic algorithm.*

Suppose there is a set of 3*N* rows of historical data (referred to as the training material) describing the state of concepts in the system. From the perspective of forecasting based on concept increments, the increments of concepts from the *i*-th iteration to $k(i+1)$ iteration form the output increment vector. In this case, the fuzzy cognitive map should indicate that, given a similar output increment vector, the concept values will change in such a way that their increments result in values at the $(i+2)$ iteration.

Let $A_i(t)$ represents the value of a concept at time *t*. Based on the specification of the training material described above, let's consider triples of rows: $A_i(t)$, $A_i(t+1)$, $A_i(t+2)$.

Define $x_i=\dfrac{A_i(t+1)-A_i(t)}{A_i(t)}$, $y_i=\dfrac{A_i(t+2)-A_i(t)}{A_i(t)}$, *x* – the output increment vectors, *y* – the resulting increment vectors.

To solve the training task, a genetic algorithm is proposed. A chromosome is represented as a one-dimensional array of values corresponding to the two-dimensional weight matrix of the fuzzy cognitive map. Each value in this array is called a gene. The main steps of the algorithm are:

1. For all non-zero weights of the initial map, a new non-zero weight value is assigned as a small random value. The initial non-zero weights are determined by the expert (any non-zero value can be assigned; its sole purpose is to indicate that, in the expert's opinion, a causal relationship exists between the two selected concepts).

2. Step 1 is repeated *Population Size* times. Thus, an initial population of Random solutions are formed.

3. The fitness function is calculated for each chromosome (the type of fitness function will be discussed later).

4. A pool of parents is determined using the "roulette wheel" selection method.

5. "Elite individuals" are added to the parent pool. In genetic algorithms, elite individuals are those that have shown the best fitness values over the last few generations (one elite individual from each generation).

6. Crossover occurs among chromosomes in the parent pool. The crossover between chromosomes A and B is performed as follows. The crossover point is determined randomly. Let's denote it as A_{l+} a segment of the chromosome A, consisting of genes located starting from l, and A_{l-} – the segment of the chromosome located before l. Then the result of the crossover will be two chromosomes: $A_{l-}B_{l+}$ and $B_{l-}A_{l+}$. The probability of crossover is predefined. If crossover does not occur, both parent chromosomes are transferred unchanged to the offspring population.

7. A new population is formed from the offspring obtained in Step 6. The size of this population is identical to the size of the population in the previous stage of the algorithm.

8. Mutations occur in the offspring population. During mutation, a random gene is selected and replaced with a new random value. The probability of mutation is predefined. If no mutation occurs, the chromosome proceeds to the next iteration of the algorithm unchanged.

9. The following parameters for the generation are determined: elite individual (the individual with the highest fitness value) retained to preserve its genetic makeup; average fitness value (this is calculated for the population and is used only to evaluate the convergence of the algorithm); fitness value of the elite individual.

10. If the fitness value of the elite individual exceeds a predefined maximum fitness value, the algorithm stops, and the selected chromosome is decomposed into the adjacency matrix of the fuzzy cognitive map (training is considered complete). Otherwise, the algorithm returns to Step 3.

End.

A methodology for evaluation and prediction using fuzzy cognitive maps has been developed. The concept of elite individuals was introduced into the algorithm to accelerate its convergence. The number of elite individuals was set to 60, while the population size was 100. Thus, after the 60th generation, only 40 chromosomes from the current population have a chance to undergo crossover, while the rest populate the elite gene pool inherited from previous populations.

This high mutation probability, which is typically uncharacteristic of genetic algorithms, is justified in this case as it introduces genetic diversity into the population. Furthermore, the use of an elite gene pool mitigates the risk of irrevocably losing beneficial genes from previous generations.

The transient processes of different systems may converge over a specific interval. Therefore, increasing the observation interval for the system's output and the measurement frequency can improve the effectiveness of finding a solution. On the other hand, this may be attributed to the presence of numerous local optima and a relatively strong attraction zone.

The proposed methodology, unlike existing ones:

– accounts for the degree of uncertainty in information about the state of a dynamic object and the noise in the output data regarding its state;

– creates a multilevel and interconnected description of hierarchical objects;

– enhances decision-making speed in assessing object states by searching for solutions using population individuals;

– addresses the issue of reaching a global optimum.

Advantages of this research include:

– consideration of the degree of uncertainty about the object's state during calculations;

– accounting for data noise resulting from distortion in the information about the object's state;

– reduction in computational costs when evaluating object states;

– the ability to perform calculations with input data of different natures and units of measurement.

Disadvantages of this research include: the need for significant computational resources and time to perform calculations.

The proposed methodology is advisable for implementation in specialized software used for analyzing the states of complex technical systems and making management decisions.

Future research should focus on further improving this methodology to account for a larger number of factors during state analysis.

6.3 DEVELOPMENT OF A METHOD FOR COGNITIVE REPRESENTATION OF THE STATE OF COMPLEX HIERARCHICAL SYSTEMS

Currently, there are no unified principles for constructing cognitive representations that convey sufficient information in a concise and user-accessible form to make appropriate management decisions [1–3].

Typically, representations are created individually, considering the specific applied domain and interpreted by an expert (or a group of experts) based on accumulated knowledge. Multidimensional data can be transformed into cognitive graphical representations using computational tools in the form of integral functional profiles, which reflect the states of complex hierarchical systems [3–6].

Existing mathematical methods for analyzing and visualizing multidimensional data are poorly suited to real-time dynamic systems and lack sufficient versatility, hindering their widespread adoption.

The aim of the study is to develop a method for cognitive representation of the states of complex hierarchical systems with the following features [4–9]:

– the systems operate in real-time;

– various scales and ranges of controlled parameter changes are applied;

– data stream interruptions and system failures may occur [2].

To visualize the current state in such systems, a cognitive graphical representation is required that meets the following requirements [3, 4]:

1. A mathematical framework for transforming feature space into image space:

– expressiveness of images to accelerate experts' understanding of the current situation;

– unambiguous and accurate representation of situation classes: "normal", "anomaly", "critical", "no info" (absence of information);

– capability to represent the state of complex hierarchical systems as a whole and the states of their individual subsystems at all hierarchy levels.

2. Ability to display parameters with an indication of the deviation level from the average within an acceptable operating range.

3. A unified formalism for describing relationships important for decision-making in high-dimensional symbolic space.

Considering the methods for constructing cognitive representations, multilevel approaches to presenting situations are the most effective for monitoring hierarchical systems. In practical applications, the number of levels typically does not exceed three, as increasing them complicates perception and reduces the efficiency of state analysis. The results of the review of methods for representing the states of complex hierarchical objects are summarized in **Table 6.1** [10–19].

Analysis of **Table 6.1** shows that the cognitive representation of information using fractals best meets the formulated requirements. However, it has several disadvantages that reduce its ergonomic qualities, namely: a large number of small details, a low permissible embedding depth, and the absence of numerical information on the cognitive image slide.

As a basis for transforming the parameter space into the space of graphical representations, the method of integral contour representation of a standard epicycloid is adopted:

$$\begin{cases} x = (R+d)\cos\varphi - d\cos(m+1)\varphi; \\ y = (R+d)\sin\varphi - d\sin(m+1)\varphi, \end{cases} \tag{6.13}$$

and standard hypocycloids:

$$\begin{cases} x = (R-d)\cos\varphi + d\cos(m-1)\varphi; \\ y = (R-d)\sin\varphi - d\sin(m-1)\varphi, \end{cases} \tag{6.14}$$

where R is the radius of the stationary circle; r is the radius of the rolling circle; $d=r$ is the distance of point M from the center C of the rolling circle; φ is the parameter describing the angle between the line segment connecting the circle centers and the OX; m is an integer defined a, $m=R/r$.

Table 6.1 Overview of methods for representing the state of complex hierarchical objects

No.	Name	Purpose	Types of systems		Deviation level	Univer-sality
			Dy-namic	Complex Hierarchies		
1.	*n*-Simplex	Pattern recognition	–	+	+	+
2.	Meta	Pattern recognition	–	+	+	–
3.	Star	Graphical representation of multidimensional data	+	–	–	+
4.	Botanical tree	Visualization of complex hierarchies	–	+	–	–
5.	Novosyolov fractal	Diagnostics and visualization for complex technical objects	+	+	–	+
6.	Image for nuclear power plant operators	Display of anomalies and current situations in complex systems	+	+	+	–
7.	Color coding based on evaluation function	Assessment of object state compliance at a given time	+	–	+	–
8.	Large-scale electri-cal network image	Diagnostics and visualization for complex technical objects	+	+	+	–
9.	Drum-separator image	Decision support for nuclear power plant operators	+	–	+	–
10.	Shoke integral	Selection of constraints on parameters of fuzzy aggrega-tion operators for interrelated criteria	–	–	+	–
11.	Fuzzy cognitive maps	Analyst decision support	–	–	+	–

When the rolling circle rotates around the stationary circle $O(x_0, y_0)$, by an angle multiple of $2\pi r/R$, the epicycloid (or hypocycloid) overlaps with itself. Both the epicycloid and hypocycloid consist of m congruent branches. Let j – the branch number, and if $j = 1,\dots,m$, then the parametric equations for the j-th congruent branch of the epicycloid are derived from expressions (6.13) and (6.14):

$$\begin{cases} x = x_0 + (R+d)\cos\psi - d\cos(m+1)\psi; \\ y = y_0 + (R+d)\sin\psi - d\sin(m+1)\psi, \end{cases} \tag{6.15}$$

and for the hypocycloid:

$$\begin{cases} x = x_0 + (R-d)\cos\psi + d\cos(m-1)\psi; \\ y = y_0 + (R-d)\sin\psi - d\sin(m-1)\psi, \end{cases} \tag{6.16}$$

where x_0 and y_0 – the coordinates of the center of the stationary circle, and $\psi \in \left[j\alpha, (j+1)\alpha\right]$, $\alpha = 2\pi/m$. From equations (6.15) and (6.16), let's obtain the generalized formula for the branches of the epicycloid and hypocycloid:

$$\begin{cases} x = x_0 + (R + \xi d)\cos\psi - \xi d\cos(m + 1\xi)\psi; \\ y = y_0 + (R + \xi d)\sin\psi - d\sin(m + 1\xi)\psi, \end{cases} \tag{6.17}$$

where $\xi = 1$ for the epicycloid and $\xi = -1$ for the hypocycloid.

Let the hierarchical system be characterized by a set of parameters $Z = \{z_1, z_2, \ldots, z_j, \ldots, z_m\}$. It is proposed to use formula (6.17) as the basis for integral contour representation to address the problem of detecting parameter values z_j that exceed the permissible range $\left[z_{j_{\min}}, z_{j_{\max}}\right]$. Let d equal the value of parameter z_j, normalized to the interval $[-\delta, \delta]$. For normalization, let's use the formula:

$$\bar{z}_j = \delta\left(2\left(\frac{z_j - z_{j_{\min}}}{z_{j_{\max}} - z_{j_{\min}}}\right) - 1\right), \tag{6.18}$$

where $\delta = R/2\Phi^2$, Φ is the golden ratio coefficient.

If the parameter z_j equals the mean value within the permissible range, then $d = \bar{z}_j = 0$ and the curve described by (6.17) becomes a part of a circle. If $\bar{z}_j < 0$, the curve (6.17) forms a congruent branch of a hypocycloid. If $\bar{z}_j > 0$, the curve (6.5) forms a congruent branch of an epicycloid.

Substituting in formula (6.17) ξd on $\bar{z}_j$, d on $\left|\bar{z}_j\right|$, the type of the curve (6.17) will be determined by the deviation of the controlled parameter from the mean value within the permissible range:

$$\begin{cases} x = x_0 + (R + \bar{z}_j)\cos\psi - \bar{z}_j\cos(m + \eta)\psi; \\ y = y_0 + (R + \bar{z}_j)\sin\psi - \left|\bar{z}_j\right|\sin(m + \eta)\psi, \end{cases} \tag{6.18}$$

where $\eta = \begin{cases} 1, & \text{if} \quad \bar{z}_j > 0; \\ -1, & \text{if} \quad \bar{z}_j < 0; \\ 0, & \text{if} \quad \bar{z}_j = 0. \end{cases}$

In cases where $z_j = q_1 z_{j_{\min}}$, $q_1 > 1$ or when $z_j = q_2 z_{j_{\max}}$, $q_2 > 1$, $\{q_1, q_2\} \in R$, the d, the corresponding $\left|\bar{z}_j\right|$, will exceed r, and the curve (6.18) will form a branch of an elongated epicycloid or hypocycloid, resulting in "loops". To eliminate loops, replace the epicycloid branches with elliptical arcs if $\bar{z}_j > r$, replace the hypocycloid branches with hyperbolic branches if $\bar{z}_j < -r$. The formula (6.18) is adjusted based on these refinements:

$$\begin{cases} x = x_0 + \left(R + \bar{z}_j\right)\cos\psi - \bar{z}_j\cos\left(m+\eta\right)\psi; \\ y = y_0 + \left(R + \bar{z}_j\right)\sin\psi - \left|\bar{z}_j\right|\sin\left(m+\eta\right)\psi, \end{cases}$$

if $\quad -r \le \bar{z}_j \le r$;

$$\begin{cases} x = x_0^e + a^e\cos\beta\cos\tau - b^e\sin\beta\sin\tau; \\ y = y_0^e + a^e\cos\beta\sin\tau + b^e\sin\beta\cos\tau, \end{cases}$$

if $\quad \bar{z}_j > r$;

$$\begin{cases} x = x_0 + a^h\cosh\theta\cos\tau - b^h\sinh\theta\sin\tau; \\ y = y_0 + a^h\cosh\theta\sin\tau + b^h\sinh\theta\cos\tau, \end{cases}$$

if $\quad \bar{z}_j < -r$, (6.19)

where $\tau = j\alpha + \alpha/2$ is the angle of rotation of the ellipse or hyperbola around the point $O\left(x_0, y_0\right)$; $x_0^e = x_0 + R\cos\alpha/2\cos\tau$, $y_0^e = y_0 + R\cos\alpha/2\cos\tau$, $a^e = 2\bar{z}_j + R - R\cos\alpha/2$ is the major semi-axis of the ellipse; $b^e = R\sin\alpha/2$ is the minor semi-axis of the ellipse; $a^h = R + 2\bar{z}_j$ is the real semi-axis of the hyperbola; $b^h = \dfrac{a^h R\sin\dfrac{\alpha}{2}}{\sqrt{\left(R\cos\dfrac{\alpha}{2}\right)^2 - \left(a^h\right)^2}}$ is the imaginary semi-axis of the hyperbola, $\theta \in \left[-2\pi, 2\pi\right]$, $\beta \in \left[-\pi/2, \pi/2\right]$.

The proposed method, unlike existing ones:

– creates a visual, multilevel, and interconnected description of a hierarchical system;

– improves the efficiency of decision-making in assessing the state of a hierarchical system;

– resolves the issue of global and local extrema when evaluating the state of a hierarchical system;

– combines graphical and numerical representations of controlled parameters for the state of the national security system;

– enables real-time visualization of the state of a hierarchical system;

– avoids the issue of loop formation during the real-time visualization of the state of a hierarchical system;

– ensures the accuracy of the hierarchical system state visualization, regardless of the number of individual components comprising the system.

Advantages of this study include:

– the ability to perform calculations with input data of different natures and measurement units;

– the ability to avoid loop formation during the visualization of the hierarchical system state;

– the creation of a visual, numerical, multilevel, and interconnected description of the hierarchical system.

Disadvantages of this study include the need for appropriate computational resources and time to perform calculations.

The proposed method is recommended for implementation in specialized software used for analyzing the state of hierarchical systems and making managerial decisions.

6.4 DEVELOPMENT OF A METHOD FOR EVALUATING COMPLEX HIERARCHICAL SYSTEMS BASED ON AN ENHANCED PARTICLE SWARM OPTIMIZATION

This study proposes a method for evaluating complex hierarchical systems using an enhanced particle swarm optimization (PSO) combined with the coordinate averaging method.

This approach allows the coordinate averaging method to shift from the random selection of trial points to utilizing the current coordinates of the particle swarm, whose collective movement is adaptive, adjusting to the characteristics of the objective function's changes. During the movement of the particle swarm, their displacement is adjusted toward the determined averaged center based on the coordinate averaging method. An additional mechanism that accelerates the convergence of the hybrid algorithm is the inclusion of several steps of the Hooke-Jeeves procedure. These steps refine the current coordinates of the best and/or worst particle in the swarm.

This study focuses primarily on "zero-order" methods, in which the values of the objective function are determined only at trial points through a computational algorithm. This approach is oriented towards solving tasks for determining the objective function of a complex dynamic process with minimal requirements for the continuity and boundedness of the objective function. For approximate estimates of the variability of the objective function, the method utilizes the maximum values obtained as the ratio of the difference in the objective function values to the distance between all pairs of trial points (a lower bound for the Lipschitz constant).

A bounded continuous function is considered $f(x):\Omega \to \mathbb{R}$, where $x=(x_1,x_2,\ldots,x_n)\in\Omega\subset\mathbb{R}^n$. The set Ω represents the domain of allowable variable values and, in the simplest case, is an n-dimensional parallelepiped with given sides, $\left[x_i^{[0]}-d_i, x_i^{[0]}+d_i\right]$, $i=1,2,\ldots,n$.

The task is to find an approximate value of the global minimum f^* and at least one point x^*, where this value is achieved, with a specified permissible ε^f for the objective function values:

$$f_{\min}=\min f(x),$$
$$x\in\Omega,\ f^*-\varepsilon_f\le f_{\min}\le f^*,\ f^*=f(x^*). \tag{6.20}$$

The computational procedure for finding the approximate coordinates of the point x^* in the coordinate averaging method is based on an iterative process, which in its continuous form is expressed as [5]:

$$x_i^{[k+1]}=\left(\int\limits_{\Omega^{[k]}} x_i p_s^{[k]}(x)\,\mathrm{d}x\right)\times\iota,\ i=1,2,\ldots,n. \tag{6.21}$$

$$p_s^{[k]}(x) = \frac{P_s^{[k]}(f(x))}{\int\limits_{\Omega^{[k]}} P_s^{[k]}(f(x))\mathrm{d}x}, \tag{6.22}$$

where k is the step number of the computational process; ι is the degree of uncertainty in the state of a complex real-time dynamic system (values range from 0 to 1); $\Omega^{[k]}$ is the coordinate averaging region at step k. A sequence of continuous functions $P_s(y)$, $s=1,2,3,\ldots$, such that $\forall y \in \mathbb{R}$ and value $P_s(y) \geq 0$ and for the sequence $P_s(y)/P_z(y)$ the condition of monotonic unbounded growth holds with increasing selectivity parameter s for any fixed values y, z with the condition $y < z$. Examples of such functions include $P_s(y)$, in particular, are functions $\exp(-sy)$, s^{-y}, y^{-s}, as well as a class of functions of the form $(1-y^r)^s$ for $y \in [0,1]$, $r=1,2,3,\ldots$. For the numerical minimization examples below, the function is used $(1-y^2)^s$.

As increases s the steepness of the kernels $p_s^{[k]}$ grows, which in turn increases the weights of the coordinates corresponding to better values of the objective function (OF). In the final case, the sequence of averaged coordinates converges to the global minimum (the corresponding convergence theorem is proven in [5]).

For the numerical implementation of the coordinate averaging method, one effective way to improve the accuracy of integral computations is to sequentially increase the number of trial points $x^{(j)[k]}$, $j = 1,2,\ldots,M^{[k]}$, at the k-th stage of the iterative process, i.e. $M^{[k]} \geq M^{[k-1]}$. To avoid accidentally excluding the global minimum point, the averaging region $\Omega^{[k]}$ in this case can be considered either adaptively variable or fixed [5].

In the proposed modification of the iterative coordinate averaging algorithm, adaptive displacement of trial points is introduced. This is implemented as the movement of particles in the PSO method with an FDR (fitness-distance ratio-based PSO) modification [22]. Additionally, particles in the swarm are shifted toward the determined center of averaged coordinates, introducing a new factor of information exchange between particles and adding extra stabilization to the collective swarm search for the global minimum of the objective function (OF).

When calculating integrals in formulas (6.21) and (6.22), the summation of the values of the integrand expressions is performed over the set of trial points, taking into account the volumes of the subregions discretizing the integration domain $\Omega^{[k]}$.

In the proposed hybrid algorithm, the coordinates of the trial points are identified with the coordinates of the particle swarm, which change during the collective search for the global minimum. At the beginning of the computational process, these coordinates are initialized either at the nodes of a computational grid or generated using a random number generator with a uniform distribution over the intervals $\left[x_i^{[0]} - d_i, x_i^{[0]} + d_i\right]$, $i = 1,2,\ldots,n$. In discrete form, the relationships (6.21) and (6.22) are expressed as follows:

$$x_i^{[k+1]} = \sum_{j=1}^{M^{[k]}} x_i^{(j)[k]} p_s^{[k]}\left(x^{(j)[k]}\right) V^{(j)[k]},\ i = 1,2,\ldots,n, \tag{6.23}$$

$$p_s^{[k]}\left(x^{(j)[k]}\right)=\frac{P_s^{[k]}\left(g^{[k]}\left(x^{(j)[k]}\right)\right)}{\sum_{j=1}^{M^{[k]}}P_s^{[k]}\left(g^{[k]}\left(x^{(j)[k]}\right)\right)V^{(j)[k]}},\tag{6.24}$$

where $V^{(j)[k]}$ corresponds to the n-dimensional volume obtained by dividing the domain $\Omega^{[k]}$ into subdomains associated with the family of integration points $x^{(j)[k]}$, $j=1,2,\ldots,M^{[k]}$.

Here $g^{[k]}(x)$ are auxiliary functions that scale the objective function $f\left(x^{(j)[k]}\right)$ to the range [0, 1], defined as:

$$g^{[k]}\left(x^{(j)[k]}\right)=\frac{f\left(x^{(j)[k]}\right)-f_{\min}^{[k]}}{f_{\max}^{[k]}-f_{\min}^{[k]}},\tag{6.25}$$

$$f_{\min}^{[k]}=\min\left(f\left(x^{(1)[k]}\right),f\left(x^{(2)[k]}\right),\ldots,f\left(x^{\left(M^{[k]}\right)[k]}\right)\right),$$

$$f_{\max}^{[k]}=\max\left(f\left(x^{(1)[k]}\right),f\left(x^{(2)[k]}\right),\ldots,f\left(x^{\left(M^{[k]}\right)[k]}\right)\right).$$

Assuming that for the trial points (current coordinates of the particle swarm) in formulas (6.24) and (6.25), during approximate integration, the subdomains can be chosen such that their sizes $V^{(j)[k]}$, $j=1,2,\ldots,M^{[k]}$, are approximately equal, the computational formulas simplify to the following form:

$$x_i^{[k+1]}=\sum_{j=1}^{M^{[k]}}x_i^{(j)[k]}p_s^{[k]}\left(x^{(j)[k]}\right),\ i=1,2,\ldots,n,\tag{6.26}$$

$$p_s^{[k]}\left(x^{(j)[k]}\right)=\frac{P_s^{[k]}\left(g^{[k]}\left(x^{(j)[k]}\right)\right)}{\sum_{j=1}^{M^{[k]}}P_s^{[k]}\left(g^{[k]}\left(x^{(j)[k]}\right)\right)}.\tag{6.27}$$

It should be noted that for the coordinate averaging computational algorithm [5], the specific type of subdomains discretizing the integration domain $\Omega^{[k]}$ is not critical, thus, using the simpler relationships (6.26) and (6.27) instead of (6.24) and (6.25) for numerical implementation is entirely justified.

The adaptive adjustment of particle swarm coordinates at the $(k+1)$-th step follows the PSO scheme [10–18], with an additional component introduced for movement toward the averaged coordinate center $x^{[k]}$ for each j-th particle in the swarm, expressed as:

$$x^{(j)[k+1]}=x^{(j)[k]}+\alpha Y^{(j)[k]}+D^{(j)[k]}+U^{[0,\,\beta_0]}\otimes\left(x^{[k]}-x^{(j)[k]}\right),\tag{6.28}$$

where $Y^{(j)[k]}$ – the inertia component of the movement of the j-th particle; $D^{(j)[k]}$ – the adaptive displacement vector of the j-th particle, which is determined by three components of the random displacement of this particle [18]:

$$D^{(j)[k]} = d_1^{(j)[k]} + d_2^{(j)[k]} + d_3^{(j)[k]}, \tag{6.29}$$

where

$$\begin{aligned} d_1^{(j)[k]} &= U\left[0,\beta_1\right] \otimes \left(x_b^{(j)[k]} - x^{(j)[k]}\right), \\ d_2^{(j)[k]} &= U\left[0,\beta_2\right] \otimes \left(x_g^{(j)[k]} - x^{(j)[k]}\right), \\ d_3^{(j)[k]} &= U\left[0,\beta_3\right] \otimes \left(x^{(q(j))[k]} - x^{(j)[k]}\right). \end{aligned} \tag{6.30}$$

In formulas (6.28)–(6.30), the following notations are used:

– $x_b^{(j)[k]}$ – the best coordinates of the j-th particle over iterations, determined based on the objective function value ($d_1^{(j)[k]}$ – represents the cognitive component of the particle displacement);

– $x_g^{(j)[k]}$ – the coordinates of the best particle in the swarm with the minimum objective function value over k iterations ($d_2^{(j)[k]}$ – represents the social component of the particle displacement);

– $x^{(q(j))[k]}$ – coordinates of the particle with index $q(j)$, in the direction of which the objective function's rate of decrease is the greatest ($d_3^{(j)[k]}$ – a component of the objective function's variability based on the local Lipschitz constant estimate);

– $U\left[0,\beta\right]$ – a vector with components uniformly distributed random numbers in the interval $\left[0,\beta\right]$; $\otimes$ – element-wise multiplication of vectors; coefficients α, β_m, $m = 0,1,2,3$ that are tunable parameters of the hybrid computational algorithm.

Thus, the relationships (6.24)–(6.30), after specifying the type of kernels $P_s^{[k]}\left(y\right)$ with an increasing selectivity parameter and assigning specific values to the coefficients α, β_m, $m = 0,1,2,3$, fully define the hybrid computational algorithm for global optimization based on the coordinate averaging and particle swarm methods. The selection of coefficients for the numerical global optimization algorithm can be performed through meta-optimization [19], which is beyond the scope of this research.

The algorithm was run 100 times, and acceptable accuracy (on the order of 10^{-2}) in the coordinates of the best particle was achieved within 10–15 iterations, followed by the sequential concentration of particles near the global minimum. It should be noted that the method is statistical, and obtaining an "acceptable result with a certain probability" requires multiple runs of the software application with different random vector values $U\left[0,\ \beta\right]$.

Transitioning to higher dimensions of variables necessitates an exponential increase in the number of trial points or particles in the swarm. The possibility of parallel computations supports the feasibility of using multiple families of particle swarms in the algorithm.

Computational experiments minimizing function (6.29) with $n=100$ showed that if the total number of particles is not increased, the hybrid algorithm converges to one of the local minima. This occurs because the local minimum has the largest attraction basin (as reflected in the last term of formula (6.29)), which increases the likelihood that at least one particle enters this region, exerting the maximum influence on the swarm's subsequent behavior.

It is worth noting that the use of multiple families of particle swarms enables simultaneous identification of both the global minimum and local minima of the objective function in certain cases. This feature can be valuable for solving applied problems.

The proposed method, unlike existing ones:

– creates a multilevel and interconnected description of complex hierarchical real-time systems;

– improves decision-making efficiency in evaluating the state of complex hierarchical real-time systems;

– resolves the issue of reaching global and local extrema when evaluating the state of complex hierarchical real-time systems;

– enables directed search by multiple particles in a given direction, considering the degree of uncertainty;

– allows for repeated analysis of the state of complex hierarchical real-time systems.

Advantages of this study include:

– the ability to perform calculations with input data of different natures and units of measurement;

– the ability to conduct directed search by multiple particles in a given direction, considering the degree of uncertainty;

– the capability for repeated analysis of the state of complex hierarchical real-time systems.

Disadvantages of this study include the requirement for substantial computational resources and time to perform calculations.

This method is recommended for implementation in specialized software used for analyzing the state of complex hierarchical real-time systems.

CONCLUSIONS

1. An analysis of knowledge representation models was performed, and the advantages of applying production-based knowledge representation in expert systems were substantiated. The main concepts of fuzzy expert systems were outlined, based on which a formal problem statement for accelerating decision-making in the rule base of a fuzzy expert system was proposed. The stages of fuzzy logical inference were analyzed.

2. A methodology for evaluation and prediction using fuzzy cognitive maps was developed. The novelty of the proposed methodology lies in:

– accounting for a correction coefficient for the degree of uncertainty about the object's state in calculations;

– adding a correction coefficient for data noise resulting from distorted information about the object's state;

– reducing computational costs when evaluating object states;

– creating a multilevel and interconnected description of hierarchical objects;

– adjusting the object description due to changes in its current state using a genetic algorithm;

– enabling calculations with input data of different natures and measurement units.

This methodology is recommended for implementation in specialized software used for analyzing the states of complex technical systems and making management decisions.

3. A method for visualizing the states of hierarchical systems was developed. The novelty of the proposed method lies in:

– creating a visual, multilevel, and interconnected description of the hierarchical system;

– improving decision-making efficiency when evaluating the state of the hierarchical system;

– resolving the issue of reaching global and local extrema when evaluating the state of the hierarchical system;

– combining graphical and numerical representations of the controlled parameters of the hierarchical system's state;

– avoiding loop formation issues during real-time visualization of the hierarchical system's state.

4. A method for evaluating complex hierarchical systems based on enhanced particle swarm optimization (PSO) was developed. The proposed method is based on combining particle swarm optimization and coordinate averaging methods, along with modifications using multiple particle swarms and incorporating the Hooke-Jeeves procedure with corresponding correction coefficients. The novelty of the proposed method lies in:

– creating a multilevel and interconnected description of complex real-time hierarchical systems;

– improving decision-making efficiency when evaluating the states of complex real-time hierarchical systems;

– resolving the issue of reaching global and local extrema when evaluating the states of complex real-time hierarchical systems;

– enabling directed search by multiple swarm particles in a given direction, considering the degree of uncertainty;

– allowing repeated analysis of the states of complex real-time hierarchical systems.

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

CHAPTER 6

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. 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

Edited by
Svitlana Kashkevich

EDECISION SUPPORT SYSTEMS: MATHEMATICAL SUPPORT

Svitlana Kashkevich, Iraida Stanovska, Oleh Shknai, Oleksandr Lytvynenko,
Yevhen Artamonov, Andrii Veretnov, Oleksandr Gaman, Andrii Radchenko, Serhii Neronov,
Zaur Musaiev, Oleg Sova, Olena Nechyporuk, Oleksii Nalapko, Oleksiy Buyalo, Anna Lyashenko,
Olha Matsyi, Anastasiia Voznytsia, Daniil Krant, Kostiantin Radchenko, Olena Shaposhnikova,
Nataliia Saienko, Viktoriia Shevchenko, Nataliia Holeho, Taras Zaloznyi, Oleksandr Yaryzhko,
Olena Nosyryeva, Anzhela Popova, Ihor Pimonov, Dmitry Leinyk

Collective monograph

Technical editor I. Prudius
Desktop publishing T. Serhiienko

TECHNOLOGY CENTER PC®
Published in February 2025
Enlisting the subject of publishing No. 4452 – 10.12.2012
Address: Shatylova dacha str., 4, Kharkiv, Ukraine, 61165
