Date of publication MAR-31, 2025, date of current version FEB-10, 2025.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.24.1.3875

Method and Rules for Determining the Next Centralization Option in Multicomputer System Architecture

ANTONINA KASHTALIAN[1], SERGII LYSENKO[1], TETIANA KYSIL[1], ANATOLIY SACHENKO[2, 3], OLEG SAVENKO[1], BOHDAN SAVENKO[1]

[1]Department of Computer Engineering and Information Systems, Khmelnytskyi National University, Khmelnytskyi, Ukraine
[2]Reasearch Institute for Intelligent Couter systems, West Ukrainian National University, Ternopil, 46009, Ukraine\
[3]Department of Informatics and Teleinformatics, Kazimir Pulaski Radom University, Radom, 26-600, Poland

Corresponding author: Sergii Lysenko (e-mail: sprlysenko@gmail.com).

ABSTRACT The paper poses a scientific problem regarding the development of multi-computer systems that would be the basis for their use in the field of cybersecurity and information protection. One of the problematic tasks that needed to be solved was the development of a method for determining the next option for centralization in systems without user intervention in order to complicate the search for the center of the system for attackers and establish the principles of their functioning. As a result of the research, methods for synthesizing systems and systems that are designed to function in corporate networks and can change their architecture during operation, that is, are adaptive, were analyzed. According to the results of the study, insufficient detailing of the internal architecture of systems was established in terms of mechanisms that launch and implement the restructuring of systems, including the center of systems. In the analyzed works, attention is mainly focused on the migration of the center between system components. The choice of the next option for the center of systems is not detailed. Therefore, the task was set in the context of the development of the theory of distributed systems to develop a method for determining the next option for centralization in systems. The work formalized the components and elements of the systems, the connections between them, the operating environment of the systems and their centers, and based on them, rules were developed for selecting the next centralization option. The obtained rules became the basis of the developed method for determining the next centralization option in systems during their restructuring without the involvement of an administrator. A feature of the developed method is the avoidance of complete or significant partial search when selecting a centralization option. To confirm the effectiveness of the proposed solution, an experimental system was developed and a study of centralization options was conducted with it. Also, machine modeling of such a system was carried out. The obtained theoretical and experimental results showed their convergence and confirmed the feasibility of using the developed method. The directions of further research are the development of a systems controller for selecting one and approving the solution options developed in the centers of the systems.

KEYWORDS multicomputer systems; deception systems; centralization; bait; traps.

I. INTRODUCTION

A. MULTI-COMPUTER SYSTEMS FOR DETECTING MALICIOUS SOFTWARE

Multi-computer systems for detecting malicious software (MS) and computer attacks (CA) in corporate networks [1] can contain baits and traps. This improves the effectiveness of their use. One of the most problematic parts of multi-computer systems for this purpose is the center. Attackers, studying systems for warning, countermeasures, and detection of MS and CA, direct their efforts and resources to search for and disable the centers of such systems. Therefore, the restructuring of the center of system and its migration during the operation of the systems would significantly improve the resistance of such systems to the effects of attackers. In addition, the center of such systems must then determine its next implementation option and its placement in certain components. There can be a lot of such options. And, accordingly, the development of methods for determining the next option of centralization in multi-computer systems is an urgent task, which should include the development of mechanisms for implementation in the architecture of systems and their further independent operation without administrator intervention.

Determining the next centralization option in the architecture of multicomputer systems requires evaluating possible options and choosing the best option, taking into account the state of the system and the corporate network environment, external influences, and the level of resource load. But there are a lot of options under consideration, and evaluating each of them takes a lot of time, which in a constantly changing environment can lead to a loss of relevance of the choice result. Therefore, the current task is to develop a method for determining the next centralization option in the architecture of class $\mathfrak{S}$ systems, for which the principles of synthesis and their models are given in [1]. Under class $\mathfrak{S}$ systems, we will consider multicomputer systems of antivirus combined baits and traps, the peculiarity of which is their synthesis in such a way as to make it impossible for attackers to detect their center. To achieve this goal, the architecture of such systems includes a model of dividing the center into two parts. The first part includes the center of system, in which the influences in the system are processed and solutions are developed. In particular, several solutions for one specific task. The second part includes a separate controller. It is responsible for approving one of the proposed solutions based on previous experience. It must be designed in such a way that it can be implemented in the architecture of class $\mathfrak{S}$ systems. That is, according to the steps of this method, the center of system must determine potential options for centralization. Resource costs and time for their determination must be minimal. And these options must be effective for subsequent steps of the system. The method must take into account the constant variability of the operating environment, a complete or significant partial search of all options must not be used. It is also necessary that previous experience in choosing previous options be taken into account. Experience should take into account the effectiveness of the system's operation with a certain center, that is, an assessment of the choice of the center of system option based on the results of the previous time the system operated with such a center.

B. A PREVIOUS WORK

In [1], a model of multicomputer antivirus combined bait and trap systems was developed. The model was based on the characteristic features inherent in distributed systems and can be combined during their synthesis. As a result, the resulting model of synthesis of such systems became the basis for the development of systems with different characteristic properties, and due to the combination of various characteristic properties, the number of variants of such systems became large. Directly, the synthesis of such systems involves changing the architecture of the system independently during its operation without user intervention. That is, the model of systems obtained in [1] became the basis for the synthesis of systems that can change their architecture and at the same time receive other characteristic properties as a result of the change. The use of such systems is very relevant for the field of cybersecurity and information protection. Systems built according to the model obtained in [1] can be the basis for the creation of certain specialized systems on their basis: systems with bait and traps; deception systems; systems for warning, detecting and countering of malware, etc.

Partial cases of the solution proposed in [1] are partially centralized systems for preventing, malware and cyberattack detecting, the principles of synthesis of which are given in [2, 3]. Systems that are developed taking into account the methods from [2, 3] can change their architecture, including the decision-making center of the system, in the process of their operation, but within the framework of an exclusively partially centralized type of architecture. The results of the application of such systems for detecting worm viruses confirmed the effectiveness of the application of the systems synthesized in this way compared to systems with a traditional type of unchangeable architecture.

II. RELATED WORKS

Let us consider scientific works that present research on systems that operate in computer networks and allow restructuring of their architecture in the process of their operation due to the occurrence of events related to internal and external influences. In addition, systems aimed specifically at preventing, malware detection, deception and systems with traps and decoys. Such systems have a certain specificity and, therefore, they are attributed to a certain class of systems, the synthesis of which requires the use of such an architecture that would ensure their resistance to malicious influences.

To ensure cyber deception [4, 5] against attackers, malware and computer attack detection systems must have reconfigurable strategies, thus providing adaptability, proactive protection, etc.

Modern protection systems must be intelligent, adaptive and able to outrun attackers. The work [6] presents a high-level architecture of a protection system that uses cyberdeception to ensure the stability and survivability of the system in the presence of attacks, errors and other incidents

There is no single deception strategy that fits all target system configurations and goals. The work [7] considers an approach for the implementation of active protection using means of deception for distributed systems. A prototype called KAGE has been developed that uses software-defined networking and virtualization to create an alternative, flexible environment in which deception is performed, allowing attackers to be captured and manipulated. The deception strategy depends on the goals of the distributed system, the services used and the configuration of the target system and the complexity of the environment. In [8], a protection system with a centralized approach based on decoys with software-defined switching. The work [9] presents the developed architecture of the HoneyProxy decoy system, which is based on a software-configured network, solves the essential problems of existing decoys, namely, it prevents targeted attacks on decoys, the spread of malicious programs in decoy networks, and the lack of switching to decoys. In [10], a proactive deception system is proposed, which consists of traps of various types, a network system of decoys, and a security operation center. In [11], to ensure dynamic configuration and reduce the effectiveness of continuous reconnaissance attacks by attackers, an advanced system for protecting moving targets based on software-defined networks was developed, which uses the topology of a virtual network to confuse the target network. In [12], instead of adding separate decoy systems to the corporate network, the target systems themselves can be equipped with tools to provide active protection, which reduces costs and complexity, and at the same time provides the attacker with more realistic targets.

In [13], a flexible virtual decoy network control system was developed, which is dynamically created, configured, and

deployed with low- and high-interaction decoys that emulate several operating systems.

The use of containers is effective in creating a flexible infrastructure for baits [14]. In [15], a framework that uses containerization methods and is designed to dynamically create decoy networks provides a deceptive environment for an attacker. In [16], a complex HoneyFactory architecture is proposed, which consists of five modules that generates bait networks using secure containers.

Developing adaptive cyberdeception techniques in real networks is extremely challenging due to the significant effort required to implement the core network infrastructure configuration functions required to support proactive deception, which includes real-time analysis, planning, and deployment of decoy resources. In [17], an active cyberdeception framework was developed, which has an extended API and synthesis mechanisms for the development of defenses with deceptive objects and allows observing the attacker's actions, creating deception strategies, and deploying them by automatically managing the network configuration. The work [18] investigated the integration of autonomous computing with the ideas of game theory and behavioral properties to create a system of adaptive cyber protection.

In [19], an adaptive cyber deception system is presented, which generates unique network representations of a virtual network, which does not display the physical network configuration to each host of a corporate network and changes the appearance of the network of hosts in real time, which prevents the intelligence of nodes compromised by an attacker.

The work [20] presents the developed technology of cyberdeception, which involves the integration of the strategy of deception into the working environment. Leveraged market-leading cyber deception solutions to deploy deception as a defense strategy in a defense tool environment.

In [21], the developed deception system is presented, which contains a deception network with a configuration identical to the target working network, to which traffic is redirected, which allows analyzing the tactics and methods of attackers and minimizes data compromise. The proposed technique can be applied to different configurations of physical/virtual/combined networks.

In work [22], effective cybercheating, which includes both active and passive methods, is considered. Passive deception tools use infrastructure and decoy systems to detect attacker intelligence and attacks. Because information system exploration is an attacker's first steps in the process of attacking an information system, its detection enables proactive defenses to quickly identify malicious actions and take action. Active cyber deception tools employ deception strategies and take actions in response to attackers' actions, predict attackers' behavior, and prevent successful attacks.

In [23], an active protection system based on decoy technology with a high level of interaction and a modular design that separates the decoy environment from a central node that manages the addition, removal, modification of decoys, making them easy to maintain and update.

In [24], an approach is proposed for determining deception tactics during software development, which are implemented by a set of deception components integrated with system components. In work [25], the process of including deception tactics in the early stages of development includes three stages: system modeling taking into account the subject area, security modeling taking into account threat models and security problems from the perspective of the attacker, and modeling cyber deception tactics

In control systems, to deceive attackers, it is necessary to use deception with the possibility of modeling and imitating physical processes. The attributes of cyber-physical deception and the architecture of a system with these attributes are considered in [26]. In [27], a 'Decepti-SCADA' cyber deception framework is presented, containing SCADA compatible decoys that can be generated and easily deployed in a critical infrastructure environment. Due to such a deployment of the system, the cyber security of the network is improved, confusing and distracting a potential attacker. The work [28] presents the developed protection at many levels of the cyber-physical system, which is important for countering experienced attackers. A two-level cyberdeception model is proposed, which covers the dynamic non-cooperative interaction between the attacker and the defense tool under the conditions of incomplete information, and extends the deception action space of the defense tool at the application and network levels, leaving the attacker uncertain about the true type of the system. In [29], the HoneyPhy structure for decoys of cyber-physical systems, which adapts to the behavior of these systems and devices that are part of the systems, thanks to which it is possible to create decoys for complex cyber-physical systems. The use of artificial intelligence is promising for the creation of deceptive systems. In work [30], modern deception systems use more accurate methods of recognizing malicious activity based on the technologies of user behavioral analytics, big data, and artificial intelligence. In [31], a complex deception framework is proposed, which has several levels designed to implement and support deception mechanisms, which ensures the use of artificial intelligence methods at all stages of system protection, namely detection and response to malicious actions.

Intelligent methods of improving the functioning of decoy mechanisms to prevent detection by intruders are considered in [32], and a model of automatic identification with group functions of the application, network, and system levels is proposed. Intelligent cyber fraud systems are able to dynamically plan a fraud strategy and effectively implement cyber fraud measures. The work [33] presents a prototype of a framework that allows to simplify the development of cyber deception tools for integration with intelligent agents. In work [34], a web system of cyber deception with a high level of interaction is proposed, which consists of a hybrid deep learning attack classifier.

In [35], the concept of designing a system for monitoring equipment is presented. The key element of the work is the development of an immune protocol for message exchange, archiving and self-diagnosis of all system components. This allows us to take into account the reliability of protocols for maintaining communication in distributed systems. In [36], a new game-theoretic framework for designing deception mechanisms in systems is presented. In [37], the risks of losing control over systems built using artificial intelligence are presented. The main threats to them are tools that use artificial intelligence technologies. In [38], deception methods are classified, which can be used in the development of systems resistant to such threats. In [39], a conceptual model of hybrid threats is developed, which includes deception methods. In [40], a moving target is developed to improve the complexity of systems. Such approaches make the system less homogeneous, static, and deterministic, which makes it difficult for attackers to understand the principles of its

functioning. A defensive deception tactic is presented. It introduces uncertainty for attackers and increases their training costs. This reduces the probability of successful attacks. To achieve the goal of countering attackers, computational intelligence is used. In [41], deception technologies are presented to mitigate attacks on a virtual local area network. In [42], decoys are analyzed. Representations through structural traps are developed for them.

Thus, from the analysis of existing methods of synthesis of systems that can restructure their architecture in the process of their functioning and which are aimed at their application in the field of cybersecurity and information protection, it was established that the principles, methods, algorithms and strategies by which the restructuring of the architecture of systems can be carried out are not detailed. Also, the restructuring of the center of the system itself, in particular the joint restructuring of the architecture of systems together with the center, is not sufficiently presented. Most of the works [] are aimed at presenting methods for detecting systems of a certain type, and not at presenting the internal structure of such systems. Therefore, the study of the restructuring of the architecture of multi-computer systems and their centers in the process of functioning is an urgent task.

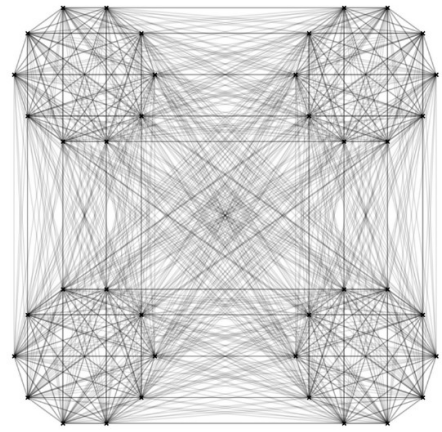
III. METHODS AND MATERIALS

A. CENTRALIZATION OPTIONS

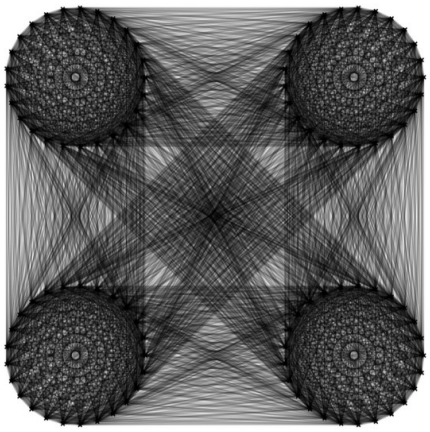
The aim of the work is to develop a method for determining the next centralization option in class $\mathfrak{S}$ systems. To achieve the goal, we detail and formalize the actions of the system for determining the next center option and the indicators that will influence its choice.

Let us divide all the options for centralization in the architecture of class S systems into four types (centralized, partially centralized, partially decentralized, decentralized) into separate classes and represent each class by a polygon. The number of vertices in the polygon should correspond to the number of options for centralization. Let us establish connections between all the vertices in the polygons and between them. As a result, we will obtain a model of options for centralization according to the states represented by the vertices. Each vertex of the polygon will correspond exclusively to one centralization option in the class $\mathfrak{S}$ architecture. The established connections between the vertices are responsible for the transitions between the centralization options. Since connections are established between any pair of vertices, the transition is possible from any centralization option to any option. But taking into account the features of the current state in the system, not all options will be equivalent for transitioning to them, therefore, rules are needed according to which promising options for transition will be determined. Fig. 1 shows four polygons and all their vertices are connected to each other. Fig. 1 a) shows a variant with four 10-gons, and Fig. 1 b) shows a 25-gon. Polygons can have a different number of vertices for each class.

Thus, there are many options for moving from one centralization option to another. The center of the system could choose the next centralization option according to one of the strategies. For example, choose from the same class of centralization types, necessarily from another class of centralization options, etc. But using such an approach to choosing the next centralization option in the system for the considered type of class $\mathfrak{S}$ systems is not suitable, since the behavior of such systems and their reaction to influences must be unpredictable and such that the attacker cannot study it. In general, the response must be effective, that is, a centralization option, and, at the same time, be polymorphic. Under the polymorphic response of the system, we will define the response of the system, which, under the same influences, will provide effective responses to influences, in particular those that are close or identical in content or essence, but the steps from which the response will be formed, and their sequence must be different. That is, for each identical impact, which is repeated with a certain periodicity or after a certain time, the system will respond to the same impacts, the steps to achieve which will be different.



a) 10-gons



b) 25-gons

Figure 1. Centralization states in the architecture of class $\mathfrak{S}$ systems.

Among the defining properties for class $\mathfrak{S}$ systems, a property such as $\mathfrak{B}_2$ was highlighted in [1], which characterizes the types and number of centers in the system architecture. All system variants will be considered exclusively taking into account the property $\mathfrak{B}_2$. When synthesizing [1] multicomputer antivirus combined bait and trap systems according to various characteristic properties and principles, it is necessary to take into account the features of centralization, that is, the architecture and functional capabilities of the centers of such systems. The effectiveness of the functioning of multicomputer antivirus combined bait and trap systems depends on the organization and features of the functioning of the center in the architecture of such systems, since this affects communication between nodes. The nodes of the system are distributed, then

the time for making decisions and sending relevant messages are important characteristics. In addition, hiding the center of multi-computer systems of antivirus combined baits and traps to avoid its detection by attackers operating from outside or from inside the perimeter of the corporate network is an important characteristic and capability of class $\mathfrak{S}$ systems. Therefore, it is necessary at the architectural level when synthesizing class $\mathfrak{S}$ systems to ensure their centralization in such a way that the systems independently hide their center, as well as to ensure effective interaction between their nodes and quick decision-making, and to maintain the integrity of the system during operation. We will specify various characteristic properties by sets of characteristic properties: type of centralization; distribution of center of system s between components; presence of the center of system in disconnected parts; features of connections between components when it is distributed between available components in the switched components in different nodes of the corporate network; hierarchy of center of system components; direction of message transmission between system components according to established connections of message transmission options with different types of communication; message transmission options with different types of communication; presence of distributed parts of the center of system in components in active and inactive states; distribution of the center of system by its parts; mixed organization of the center. of elements. These characteristic properties will characterize the features of the center organization. We will specify them as follows:

$$\mathrm{M}_{\mathfrak{B}_2,\mathrm{centr},\mathrm{v_k}} = \left\{\mathrm{m}_{\mathfrak{B}_2,\mathrm{centr},\mathrm{v_k},1}, \mathrm{m}_{\mathfrak{B}_2,\mathrm{centr},\mathrm{v_k},2}, \dots, \mathrm{m}_{\mathfrak{B}_2,\mathrm{centr},\mathrm{v_k},\mathrm{N}_{\mathrm{M}_{\mathfrak{B}_2,\mathrm{centr},\mathrm{v_k}}}}\right\}, \quad (1)$$

where $\mathrm{k} = 1,2,\dots,10$; k – number of sets of characteristic properties $\mathrm{M}_{\mathfrak{B}_2,\mathrm{centr},\mathrm{v_k}}$; element $\mathrm{m}_{\mathfrak{B}_2,\mathrm{centr},\mathrm{v_k},\mathrm{j}}$ reflects one characteristic property of class systems $\mathfrak{S}$ in the k-th set $\mathrm{M}_{\mathfrak{B}_2,\mathrm{centr},\mathrm{v_k}}$; $\mathrm{j} = 1,2,\dots,\mathrm{N}_{\mathrm{M}_{\mathfrak{B}_2,\mathrm{centr},\mathrm{v_k}}}$; $\mathrm{N}_{\mathrm{M}_{\mathfrak{B}_2,\mathrm{centr},\mathrm{v_k}}}$ – number of elements in a set $\mathrm{M}_{\mathfrak{B}_2,\mathrm{centr},\mathrm{v_k}}$.

Such a combination of sets and their elements among themselves will provide a reflection of the features of the center of system by the types of its architecture, which must be used in the operation of class $\mathfrak{S}$ systems in order to hide the center of system and confuse attackers. In fact, being in one of the possible centralization options defined by the center of system, the system is in a given state that characterizes its architecture at the current moment in time. During the operation of the system, it can independently change the centralization options, moving to another state.

Let us specify the rules for forming centralization options for multicomputer systems by a set of rules:

$$M_{Pr} = \left\{m_{Pr,1}, m_{Pr,2}, \dots, m_{Pr,N_{M_{Pr}}}\right\}, \quad (2)$$

where $m_{Pr,i}$ – i-th rule for choosing the centralization option in multi-computer systems; $i = 1,2,\dots,N_{M_{Pr}}$; $N_{M_{Pr}}$ – number of rules in the set of rules M_{Pr}.

The rules given by the set M_{Pr} will determine the conditions under which the transition from state to state will be allowed. In the absence of a set of rules that can be fulfilled, the transition from state to state will be prohibited. This is necessary in the context of the fact that not all states can be transitioned to all states.

The state of the system at the current time is given by the vector $\mathrm{V}_{\mathrm{Pr,i}}$ ($\mathrm{i} = 1,2,\dots,\mathrm{N}_{\mathrm{V}_{\mathrm{Pr}}}$; $\mathrm{N}_{\mathrm{V}_{\mathrm{Pr}}}$ is the number of states of the system). The values of the elements of the sets from formula (1) are given by the elements of the set {0;1}. Then, we introduce a Boolean function to reflect the activity/inactivity of the feature given by the element of the sets from formula (1), as follows:

$$\mathrm{F}_{\mathrm{Pr}}\left(\mathrm{m}_{\mathfrak{B}_2,\mathrm{centr},\mathrm{v_l},\mathrm{m}},\mathrm{i}\right) = \begin{cases} 0, \text{if element is absent;} \\ 1, \text{ if element is present,} \end{cases} \quad (3)$$

where l – number of sets; $\mathrm{l} = 1,2,\dots 10$; m–th element in the set $\mathrm{M}_{\mathfrak{B}_2,\mathrm{centr,l}}$; i – system state number.

Thus, vector $\mathrm{V}_{\mathrm{Pr,i}}$ ($\mathrm{i} = 1,2,\dots,\mathrm{N}_{\mathrm{V}_{\mathrm{Pr}}}$; $\mathrm{N}_{\mathrm{V}_{\mathrm{Pr}}}$ – number of the system states) we define its coordinates as follows:

$$\mathrm{V}_{\mathrm{Pr,i}} = \begin{pmatrix} \mathrm{F}_{\mathrm{Pr}}\left(\mathrm{m}_{\mathfrak{B}_2,\mathrm{centr},\mathrm{v}_1,1},\mathrm{i}\right), \mathrm{F}_{\mathrm{Pr}}\left(\mathrm{m}_{\mathfrak{B}_2,\mathrm{centr},\mathrm{v_l},2},\mathrm{i}\right), \\ \dots, \\ \mathrm{F}_{\mathrm{Pr}}\left(\mathrm{m}_{\mathfrak{B}_2,\mathrm{centr},\mathrm{v}_{10},3},\mathrm{i}\right) \end{pmatrix}, \quad (4)$$

where $\mathrm{i} = 1,2,\dots,\mathrm{N}_{\mathrm{V}_{\mathrm{Pr}}}$; $\mathrm{N}_{\mathrm{V}_{\mathrm{Pr}}}$ – number of the system states; i – system state number.

Then, the matrix $M_{\mathrm{V}_{\mathrm{Pr}}}$ of the system states in the part of the organization of centralization is given as follows

$$M_{\mathrm{V}_{\mathrm{Pr}}} = \begin{pmatrix} \mathrm{V}_{\mathrm{Pr},1} \\ \mathrm{V}_{\mathrm{Pr},2} \\ \dots \\ \mathrm{V}_{\mathrm{Pr},\mathrm{N}_{\mathrm{V}_{\mathrm{Pr}}}} \end{pmatrix}, \quad (5)$$

where $\mathrm{N}_{\mathrm{V}_{\mathrm{Pr}}}$ is the number of system states; $\mathrm{V}_{\mathrm{Pr,i}}$ is the vector that specifies the system state at the current time; $\mathrm{i} = 1,2,\dots,\mathrm{N}_{\mathrm{V}_{\mathrm{Pr}}}$.

When detailing cases and, accordingly, increasing the elements of the sets, the number of system states in the part of the organization of centralization, which are reflected in the matrix $M_{\mathrm{V}_{\mathrm{Pr}}}$, can be increased, or when reducing the elements, it can be reduced.

The transition from a state to a state, which will be determined by the rules from the set of rules M_{Pr}, is given by the function $\mathrm{V}_{\mathrm{Pr,next}}$ as follows:

$$F^{M_{Pr}}: \left(\mathrm{V}_{\mathrm{Pr},current}, M_{\mathrm{V}_{\mathrm{Pr}}}, M_{Pr}, P_{Pr}\right) \rightarrow \mathrm{V}_{\mathrm{Pr,next}}, \quad (6)$$

where $\mathrm{V}_{\mathrm{Pr,current}}$ is a vector that specifies the state of the system at the current time and is present in the state matrix $M_{\mathrm{V}_{\mathrm{Pr}}}$; $M_{\mathrm{V}_{\mathrm{Pr}}}$ is a matrix of system states in the part of the organization of centralization; M_{Pr} is a set of rules; $\mathrm{V}_{\mathrm{Pr,next}}$ is a vector that specifies the next state of the system at the current time; P_{Pr} is a set of indicators that characterize the current state of the system and the processes in it.

The set P_{Pr} is defined by its elements as follows:

$$P_{Pr} = \left\{p_{Pr,1}, p_{Pr,2}, \dots, p_{Pr,\mathrm{N}_{P_{Pr}}}\right\}, \quad (7)$$

where $p_{Pr,i}$ is the i-th indicator that characterizes the current state of the system and the processes in it; i=1,2,…, $\mathrm{N}_{P_{Pr}}$; $\mathrm{N}_{P_{Pr}}$ is the number of indicators that characterize the current state of the system and the processes in it and affect the change in the centralization option in the system.

Let $p_{Pr,1}$ be the time for choosing a centralization option in the system and transitioning from one state to another when changing centralization, $p_{Pr,2}$ be the indicator of the available components in the switched-on computer stations, $p_{Pr,3}$ be the indicator of the impossibility of completing the transition to the next state and returning to the previous state; $p_{Pr,4}$ be the indicator of the system being in an emergency state, etc. An emergency state is a state that occurs in the system when certain equipment is turned off and the system components are divided

into several disconnected subsets. Then, a centralization option is selected for each of the formed subsystems. At the same time, each of the subsystems may have different centralization options. When returning from such a state, the system must also take into account the previous state in which it was before entering the emergency state. In general, for such a class of systems, it is necessary to store information about all selected states, indicators, and rules used when choosing centralization options throughout their operation. For example, it is possible to move from a certain state to another if there is a certain number of system components in the turned-on computer stations, but with a different number of components, the transition to this state may be impossible and there will be a transition to another state.

According to formula (6), the transition to the next state can occur at different time intervals of the system's operation using different rules from the set of rules M_{Pr} and at different indicators from the set of indicators P_Pr, which characterize the current state of the system and the processes in it.

Let's divide the rules from the set of rules M_{Pr} (formula (2)) into groups, to which rules with certain common features will be assigned. Let's highlight the following common features for dividing the rules into groups:

1) transition from a certain state to the next selected state;

2) transition to a certain state as a result of an emergency situation and return to the normal operating mode of the system;

3) formation of new rules by the system using the rules available in the set and their use when the system transitions from a certain state to the next selected state.

Let's introduce the matrix M_{Pr}^1 to store information about the type of rule, that is, to which group it belongs, the time of application of the rule provided that the transition is performed according to it, the number of the rule from the set of rules M_{Pr}. Let's define the matrix M_{Pr}^1 as follows:

$$M_{Pr}^1 = \begin{pmatrix} m_{Pr,1}^1 & m_{Pr,2}^1 & \dots & m_{Pr,N_{M_{Pr}^1}}^1 \\ t_{Pr,1}^1 & t_{Pr,2}^1 & \dots & t_{Pr,N_{M_{Pr}^1}}^1 \\ N_{M_{Pr},1} & N_{M_{Pr},2} & \dots & N_{M_{Pr},N_{M_{Pr}^1}} \end{pmatrix}, \quad (8)$$

where $N_{M_{Pr}^1}$ – number of successfully completed transitions between system states; $i = 1,2,\dots,N_{M_{Pr}^1}$; $m_{Pr,i}^1$ – rule group number for i–th transition; $t_{Pr,i}^1$ – current time for the completed i-th transition; $N_{M_{Pr},i}$ – number of the element from the set of rules on the i-th transition.

Matrix M_{Pr}^1 specifies information about the previous states of the system when organizing transitions, as well as about the rules used and the time of complete completion of transitions. Such information is needed by the system to make decisions about the next centralization option and in the event of emergency situations. Separate rules are required to ensure an exit from non-standard or emergency situations. If an emergency or non-standard situation for the system has arisen, then the disconnected parts of the system decide on the centralization option in parts, and during the subsequent transition to the standard operating mode of the system, the centralization option is also determined according to separate rules. Also, for example, there may be a situation of transition to a certain state when the system was unable to complete the corresponding actions, and it is no longer possible to return to the previous state as a result of an emergency or non-standard situation, which may be due to a change in the system architecture. Then, returning to the standard operating mode of the system requires rules that would ensure the establishment of a centralization option under the existing conditions. Therefore, when the system is functioning, the set of rules should be divided into groups of rules that can be applied only under certain conditions.

The third group of rules includes rules that are formed by the system itself using simpler rules, that is, they are constructed by it to select the next centralization option. Such a group of rules is introduced into the set of rules by the system itself. The need to form such a group of rules arises during the functioning of the system when the centralization options proposed from the first group of rules are repetitive or their number does not satisfy the requirements of the controller. Then, the system forms new rules by general search and, if the controller accepts them, they are included in the set of rules for the third group.

The first group of rules forms not only the rules for transitioning from one state to another in the normal mode of operation, establishing with such an action another option of centralization in the system. That is, the rules of the first group specify not only a simple combination of possible options of the elements of the sets, which are given by formula (1). Among the rules of the first group there should be rules that reject the options for centralization in the system that are impossible. For example, if the option for centralization in the system involves centralization, then it cannot be combined with the option that provides for the division into several disconnected subsets of the set of system components. Or, for example, if the system is partially decentralized, then the option with a large number of hierarchy levels in it is impossible. Thus, in the normal mode of operation of the system, the rules form a new option for centralization in the system not by simply searching through all possible options, but taking into account the impossibility of combining certain properties that are given by the elements of the sets.

The division of rules into three groups reflects the features of their application at the current moment of the system's functioning, taking into account the events that occur in the system when it performs tasks, including changes in the centralization option.

To form the rules, we detail the features of the four main centralization options. We will specify class $\mathfrak{S}$ systems by their components as follows;

$$A^{\mathfrak{S}} = \left\{A_1^{\mathfrak{S}}, A_2^{\mathfrak{S}}, \dots, A_{N_{A^{\mathfrak{S}}}}^{\mathfrak{S}}\right\}, \quad (9)$$

where $A^{\mathfrak{S}}$ – is the designation of the class system $\mathfrak{S}$; $A_i^{\mathfrak{S}}$ - the i-th component of the system $A^{\mathfrak{S}}$ of the class $\mathfrak{S}$; $i = 1,2,\dots,N_{A^{\mathfrak{S}}}$; $N_{A^{\mathfrak{S}}}$ – s the number of components in the system $A^{\mathfrak{S}}$.

Taking into account the need to present centralization options in the system, we will divide the components of the $A^{\mathfrak{S}}$ system into two subsets. The first subset includes the components that are currently active components of the center of system. The second subset includes components that are not currently components of the center of system. We assume that any components of the $A^{\mathfrak{S}}$ system can be active components of the center if they are in switched-on computer stations. Active components are components that are functioning as part of the system at the current time. We assume that components of the $A^{\mathfrak{S}}$ system can be active, but they do not necessarily belong to the center of system at the current time. Then, we define the set

of components of the $A^{\mathfrak{S}}$ system as follows:

$$A^{\mathfrak{S}} = A_1^{\mathfrak{S}} \cup A_1^{\mathfrak{S}}, \quad (10)$$

where $A_1^{\mathfrak{S}}$ is a subset of active components of the center of system ; $A_1^{\mathfrak{S}}$ is a set of system components that are not currently components of the center of system .

Let's define these two subsets with elements as follows:

$$A_1^{\mathfrak{S}} = \left\{A_{1,1}^{\mathfrak{S}}, A_{1,2}^{\mathfrak{S}}, \dots, A_{1,N_{1,A^{\mathfrak{S}}}}^{\mathfrak{S}}\right\}; \quad (11)$$
$$A_2^{\mathfrak{S}} = \left\{A_{2,1}^{\mathfrak{S}}, A_{2,2}^{\mathfrak{S}}, \dots, A_{2,N_{2,A^{\mathfrak{S}}}}^{\mathfrak{S}}\right\},$$

where $A_{1,i}^{\mathfrak{S}}$ - i-th component of the subset $A_1^{\mathfrak{S}}$ of system $A^{\mathfrak{S}}$ of class $\mathfrak{S}$; $i = 1,2, \dots, N_{1,A^{\mathfrak{S}}}$; $N_{1,A^{\mathfrak{S}}}$ – number of elements in a subset $A_1^{\mathfrak{S}}$; $A_{2,j}^{\mathfrak{S}}$ - j-та component of the subset $A_2^{\mathfrak{S}}$ of system $A^{\mathfrak{S}}$ of class $\mathfrak{S}$; $j = 1,2, \dots, N_{2,A^{\mathfrak{S}}}$; $N_{2,A^{\mathfrak{S}}}$ – number of the elements in the subset $A_2^{\mathfrak{S}}$; $N_{A^{\mathfrak{S}}} = N_{1,A^{\mathfrak{S}}} + N_{2,A^{\mathfrak{S}}}$.

The selection of criteria will be carried out based on the goal of the evaluation objective function for choosing one of the four types of centralization, which is to minimize its value. That is, the values of the objective function will indicate how effective one of the four types of architecture is depending on the number of components in the system and their activity at the current moment of time. In this approach to constructing the objective function of selection, we will take into account the number of components in the system, including those active at the current moment of time. The four types of centralization in the architecture of the system cannot be evaluated only by simple ranking among themselves at four levels. There may be cases when one of the types with a larger number of components is less effective compared to the type of architecture in which, with the same number of components, efficiency is lower, and with a smaller number of components, efficiency is better. That is, for different types of centralization, with different numbers of components, there may be intersections in the classes of their types in terms of efficiency. To evaluate the types of centralization in architecture depending on the four main types, we will introduce the following criteria: efficiency; stability; integrity; security. For analytical presentation of criteria, we will use the following indicators: time to perform architecture restructuring in terms of centralization; time to prepare decisions regarding the next centralization option in the system architecture; total number of components in the system; number of active center components in the system at the current time; number of components with the center of system at the current time; the number of components of the centre of system in the new option, which is planned to be upgraded in the next step; number of segments in the corporate network; presence of system components in the demilitarized zone of the corporate network; presence of system components in server nodes; presence of center of system functionality in nodes in the demilitarized zone; presence of center of system functionality in server nodes.

C. AN OBJECTIVE FUNCTION FOR EVALUATING THE NEXT CENTRALIZATION

Let us set the objective function for evaluating the next centralization options for choosing one of the four types of centralization options as follows:

$$\mathrm{F}_{\mathrm{kr}}^{\mathrm{centr}}\begin{pmatrix} \mathrm{f}_{1,\mathrm{kr}}^{\mathrm{centr}}\left(\mathrm{p}_{1,\mathrm{kr}}^{\mathrm{centr}}\right), \mathrm{f}_{2,\mathrm{kr}}^{\mathrm{centr}}\left(\mathrm{p}_{2,\mathrm{kr}}^{\mathrm{centr}}\right), \dots, \\ \mathrm{f}_{\mathrm{N}_{\mathrm{F}_{\mathrm{kr}}^{\mathrm{centr}},\mathrm{kr}}}^{\mathrm{centr}}\left(\mathrm{p}_{\mathrm{N}_{\mathrm{F}_{\mathrm{kr}}^{\mathrm{centr}},\mathrm{kr}}}^{\mathrm{centr}}\right), \\ \mathrm{F}_{\mathrm{var}}^{\mathrm{centr}}(\mathrm{u}), \mathrm{F}_{\mathrm{var}}^{\mathrm{centr}}(\mathrm{v}) \end{pmatrix} \rightarrow \min, \quad (12)$$

where $\mathrm{f}_{\mathrm{i,kr}}^{\mathrm{centr}}\left(\mathrm{p}_{\mathrm{i,kr}}^{\mathrm{centr}}\right)$ – is the i-th function that specifies the calculation of the value of i-th criterion; $\mathrm{p}_{\mathrm{i,kr}}^{\mathrm{centr}}$ is a vector whose coordinates are the parameters of i-th criterion and the centralization variant; $\mathrm{N}_{\mathrm{F}_{\mathrm{kr}}^{\mathrm{centr}}}$ is the number of arguments of the function $\mathrm{F}_{\mathrm{kr}}^{\mathrm{centr}}$ and the number of vectors $\mathrm{p}_{\mathrm{i,kr}}^{\mathrm{centr}}$; vector $\mathrm{p}_{\mathrm{i,kr}}^{\mathrm{centr}} = \left(\mathrm{p}_{1,\mathrm{i,kr}}^{\mathrm{centr}}, \mathrm{p}_{2,\mathrm{i,kr}}^{\mathrm{centr}}, \dots, \mathrm{p}_{\mathrm{N}_{\mathrm{p}_{\mathrm{i,kr}}^{\mathrm{centr}},\mathrm{i,kr}}}^{\mathrm{centr}}, \mathrm{V}_{\mathrm{Pr,u}}, \mathrm{V}_{\mathrm{Pr,v}}, A_{1,u}^{\mathfrak{S}}, A_{1,v}^{\mathfrak{S}}, A_{2,u}^{\mathfrak{S}}, A_{2,v}^{\mathfrak{S}}\right)$, where $\mathrm{p}_{\mathrm{m,i,kr}}^{\mathrm{centr}}$ is the value of the m-th parameter for i – that criterion with respect to v – that centralization variant in the system; $\mathrm{N}_{\mathrm{F}_{\mathrm{kr}}^{\mathrm{centr}}}$ is the number of parameters i–th criterion; u and v are the numbers of the centralization variants; u is the number of the current centralization variant in the system; v is the number of the studied centralization variant in the system after u-th number of the centralization variant; $\mathrm{V}_{\mathrm{Pr,u}}, \mathrm{V}_{\mathrm{Pr,v}}$ – vectors given by coordinates according to formula (4); $\mathrm{F}_{\mathrm{var}}^{\mathrm{centr}}(\mathrm{v})$ – function, the value of which is the number of one of the types of centralization in the system at the current time or the number of the studied type; $A_{1,u}^{\mathfrak{S}}$ – subset of active components of the center of system at u-th current number of the centralization variant; $A_{2,u}^{\mathfrak{S}}$ – set of system components that at the current time are not components of the center of system at u-th current number of the centralization variant; $A_{1,u}^{\mathfrak{S}}$ – subset of active components of the center of system for v–th number of the studied centralization variant in the system, which can be after u-th number of the centralization variant; $A_{2,u}^{\mathfrak{S}}$ is the set of system components that are not currently components of the center of system for v-th number of the centralization variant under study in the system, which may be after u-th number of the centralization variant.

Given the need for the center of system to take into account the effectiveness of previous centralization options in the system architecture, that is, to take into account the previous period of system operation in terms of using different centralization options as experience gained, we will introduce a function for evaluating previous centralization options:

$$\mathrm{F}_{\mathrm{OD}}^{\mathrm{centr}}\left(\mathrm{V}_{\mathrm{Pr,current}}, M_{\mathrm{V}_{\mathrm{Pr}}}, P'_{Pr}\right) \rightarrow [0,1], \quad (13)$$

where $\mathrm{V}_{\mathrm{Pr,current}}$ is a vector (formula (4)) that specifies the state of the system at the current time and is present in the state matrix $M_{\mathrm{V}_{\mathrm{Pr}}}$; $M_{\mathrm{V}_{\mathrm{Pr}}}$ is a state matrix (formula (5)) of the system in the part of the organization of centralization; P'_{Pr} is a set of indicators (formula (7)) that characterize the current state of the system and the processes in it at the moment before the system transition to another option of centralization.

The value of the $\mathrm{F}_{\mathrm{OD}}^{\mathrm{centr}}$ function will be evaluated by a number in the range [0,1]. The larger value of the $\mathrm{F}_{\mathrm{OD}}^{\mathrm{centr}}$ function from the two values will correspond to the better value. The arguments $\mathrm{F}_{\mathrm{OD}}^{\mathrm{centr}}\left(\mathrm{V}_{\mathrm{Pr,current}}, M_{\mathrm{V}_{\mathrm{Pr}}}, P'_{Pr}\right)$ in the $\mathrm{F}_{\mathrm{OD}}^{\mathrm{centr}}$ function will provide the results of the system's operation with a certain centralization option and the values in them will contain the values accumulated at the time of completion of the current centralization option, that is, before

the transition to the next centralization option, which will form the system's experience in terms of centralization. In fact, the F_{OD}^{centr} function is a reward function for each centralization option in the system architecture. If the centralization option has been repeated several times during a certain time of the system's operation, then we will use its arithmetic mean value for different periods of the system's operation with the same centralization option, provided that the difference between such values is less than a certain threshold value (for example, 1%). If the values differ by more than the set threshold value, then they must be evaluated by the current center of the system for increasing order, that is, for improving the functioning of the system with such a center in its next use. If so, then these values are fixed as separate values. If the 2nd value is less than the first value, then such a centralization option will not be selected by the system in the future, and the values are fixed in the system. If the F_{OD}^{centr} function evaluates centralization options in the system that have not yet been used, then its value is zero.

To determine the value of the F_{OD}^{centr} function for each centralization option in the system that has been used, we add the $F_{OD,t}^{centr}$ function, which will reflect the time of each used centralization option in the system based on the system state vector at the current time (formula (4)). Then, for each centralization option, upon completion of its use in the system, two values will be fixed: performance assessment; time the system spent in such a centralization option.

Also, we will add to the characteristic of the used centralization variant in the system such a numerical characteristic as the number of used attempts of a certain centralization variant. We will specify the definition of its value by the function $F_{OD,k}^{centr}$, the argument of which will be the number of the centralization variant in the system.

Thus, using the function of evaluating the effectiveness of previous centralization variants in the system architecture, the time of each used centralization variant and the number of used attempts of a certain centralization variant, information will be accumulated about the previous period of system operation in terms of using different centralization variants. It will specify part of the system experience together with its assessment of effectiveness and residence time.

When choosing the next centralization variant in the system, a function is needed that will determine the possibility of switching to the next centralization variant. For example, the system in the process of functioning has decided on the next centralization variant, after which it has been divided into two unrelated parts or part of the components that were planned for further use with the center functionality have been turned off, then the planned center restructuring will be impossible. Thus, this function will determine the state of the system regarding its ability to implement the transition after all transition preparation actions. It will take into account current information about the system components. Let's define it as follows:

$$F_{P}^{centr}\left(V_{Pr,current}, V_{Pr,next}, M_{V_{Pr}}, P'_{Pr}, A^{\Subset}\right) = \begin{cases} 0, \text{transition is impossible;} \\ 1, \text{transition is possible,} \end{cases} \quad (14)$$

where $V_{Pr,current}$ is a vector (formula (4)) that specifies the state of the system at the current moment of time and is present in the state matrix $M_{V_{Pr}}$; $M_{V_{Pr}}$ is a state matrix (formula (5)) of the system in the part of the centralization organization; $V_{Pr,next}$ is a vector (formula (4)) that specifies the expected next state of the system; P'_{Pr} is a set of indicators (formula (7)) that characterize the current state of the system and the processes in it at the moment before the system transitions to another centralization option; $A^{\Subset}$ is a set of system components.

To avoid complete enumeration when choosing centralization options in the architecture of class $\Subset$ systems, it is necessary to use element selection strategies to form a centralization option. Let us introduce a set of strategies $M_{str,next}^{centr} = \left\{m_{str,next,1}^{centr}, m_{str,next,2}^{centr}, \dots, m_{str,next,N_{M_{str,next}^{centr}}}^{centr}\right\}$, where $N_{M_{str,next}^{centr}}$ is the number of strategies for choosing the next centralization option. For example, the elements of the set of strategies $M_{str,next}^{centr}$ may be as follows: $m_{str,next,1}^{centr}$ – the largest value of the objective function F_{kr}^{centr} for evaluating the next centralization options from all previously considered and used options; $m_{str,next,2}^{centr}$ – the smallest value of the objective function F_{kr}^{centr} for evaluating the next centralization options from all previously considered and used options; $m_{str,next,3}^{centr}$ – a randomly generated number that is not greater than the number of all possible options; $m_{str,next,4}^{centr}$ – the value of the objective function F_{kr}^{centr} is zero; $m_{str,next,5}^{centr}$ – a centralization option that is obtained from the previous option by replacing one element in one of the sets given by formula (1); $m_{str,next,6}^{centr}$ – a centralization option that is obtained from the previous option by replacing one element in two sets given by formula (1); $m_{str,next,7}^{centr}$ – a centralization option that is obtained from the previous option by replacing one element in three sets given by formula (1); $m_{str,next,8}^{centr}$ – a centralization option that is obtained from the current option by replacing one element in four sets given by formula (1); etc. There can be many strategies for determining the next centralization option, but the optimal ones are those that provide one option without using a complete or partial search of all possible options and without searching for the largest or smallest value of certain functions, since such actions consume computational resources and time. But options using the values of the objective function can be if sorting its values for already used centralization options is ensured. An important feature when choosing the next centralization option, that is, actually constructing it, is the simultaneous specification of the option directly by the selection strategy itself, such as in the strategies $m_{str,next,5}^{centr}$ – $m_{str,next,8}^{centr}$ and typical ones. Such strategies make it possible to form a centralization option with very close characteristic properties, closer, less close and opposite.

It is necessary to take into account the number of applications of strategies in order to avoid using several or one strategy constantly. Therefore, let us introduce a function to determine the number of applications of a particular strategy as follows:

$$F_{str,K}^{centr}\left(m_{str,next,m}^{centr}\right) = k_m, \quad (15)$$

where $m_{str,next,m}^{centr}$ is the m-th element of the set of strategies $M_{str,next}^{centr}$; $m = 1,2,\dots,N_{M_{str,next}^{centr}}$; $N_{M_{str,next}^{centr}}$ is the number of strategies for choosing the next centralization option; k_m is the number of applications of a certain m - strategy for choosing a centralization option.

Taking into account the choice of strategies themselves when choosing the next option in the context of their impact on the result of the functioning of the system and its center is taken into account in the objective function. For this, the estimated

value of the functioning of the system with such a center is introduced. Therefore, we will not separately consider the choice of strategy because there will be a duplication of this indicator.

The current center of the system, when deciding to replace it with another, must prepare five options for the system controller to choose. These options may be: the closest centralization option to the current one, i.e. the option of the same architecture as the current architecture of the center of system; three options from the remaining three types of centralization in the system architecture; an option that does not contain all the elements of the sets from which the current centralization option is formed. The rest of the tasks that the system can perform are processed by the center of the system or the components of the system determined by it, and 3-5 solutions are also prepared for them regarding a specific response to the corresponding influences or periods in the functioning of the system.

D. RULES FOR DETERMINING THE NEXT CENTRALIZATION OPTION

Let us define the indicators that must be used when determining the next centralization option according to the rules, which are given by the set of rules M_{Pr} according to formula (2), in systems of class $\mathfrak{S}$, as follows:

1) division of rules into groups (g_l^{centr} , l=1,2,3) according to the result of the transition to the next selected state and the formation of new rules using the matrix M_{Pr}^1 (formula (8), which contains information about the type of rule, the time of application of the rule provided that the transition is performed according to it, the number of the rule from the set of rules M_{Pr};

2) function $F^{M_{Pr}}$ of the transition from the current state to the next state, which is defined by formula (6), with the current state fixed by the vector $V_{Pr,current}$ and the next state determined by the vector $V_{Pr,next}$;

3) sets that specify the characteristic properties of the centralization options and which are given by formula (1);

4) the state of the system, which is defined by the vector $V_{Pr,i}$ according to formula (4) and the matrix $M_{V_{Pr}}$ of the system states according to formula (5);

5) the set P_{Pr} of indicators that characterize the current state of the system and the processes in it, given by formula (7);

6) the set $A^{\mathfrak{S}}$ (formula (9)) specifying systems of class $\mathfrak{S}$ by a list of components;

7) the division of components from the set $A^{\mathfrak{S}}$ into two subsets (formulas (10), (11));

8) the objective function F_{kr}^{centr} evaluating the next centralization options (formula (12)) according to the criteria for efficiency, stability, integrity, and security of the center of system;

9) the time t of the system operation from the beginning of its launch;

10) the time $F_{OD,t}^{centr}$ of each used centralization option in the system;

11) the function F_{OD}^{centr} of the reward for each centralization option in the system architecture;

12) function $F_{OD,k}^{centr}$ of used attempts of a certain centralization option and function $F_{OD,z}^{centr}$ number of all centralization options that were in the system at time t, and the same centralization options at different times will be considered different for calculating the value of the function $F_{OD,z}^{centr}$;

13) number of active components N_A^{centr} in the system from the total number of components in which there is a center at the current moment of system operation and can be the center of the system under the next centralization option;

14) function F_P^{centr}, which reflects the impossibility of combining certain characteristic properties, i.e. which removes options from all considered options;

15) number (n_s^{centr}, s=1,…,4) of the type of centralization architecture (n_1^{centr} – centralized; n_2^{centr} – partially centralized; n_3^{centr} – partially decentralized; n_4^{centr} – decentralized);

16) the set of strategies $M_{str,next}^{centr}$ for selecting elements to form a centralization option and the number of their previous applications.

Let us specify the rules from the set of rules M_Pr (formula (2)), according to which the next centralization option in class $\mathfrak{S}$ systems [1] will be determined, taking into account the indicators specified in points 1)-16) as follows:

$m_{Pr,s} = \forall A^{\mathfrak{S}}$: $((g_1^{centr} = \text{true})$ *and* $(g_2^{centr} = \text{false})$ *and* $(g_3^{centr} = \text{false}))$ *and* $(V_{Pr,current} \neq V_{Pr,next}$: $\exists$ j $F_{Pr}(m_{\mathfrak{B}_2,centr,v_l,j}, current) \neq F_{Pr}(m_{\mathfrak{B}_2,centr,v_l,j}, next))$ *and* $(\{m_{\mathfrak{B}_2,centr,v_1,j_1}\} \cup \{m_{\mathfrak{B}_2,centr,v_2,j_2}\} \cup \{m_{\mathfrak{B}_2,centr,v_3,j_3}\} \cup \{m_{\mathfrak{B}_2,centr,v_4,j_4}\} \cup \{m_{\mathfrak{B}_2,centr,v_5,j_5}\} \cup \{m_{\mathfrak{B}_2,centr,v_6,j_6}\} \cup \{m_{\mathfrak{B}_2,centr,v_7,j_7}\} \cup \{m_{\mathfrak{B}_2,centr,v_8,j_8}\} \cup \{m_{\mathfrak{B}_2,centr,v_9,j_9}\} \cup \{m_{\mathfrak{B}_2,centr,v_{10},j_{10}}\})$ and $(V_{Pr,current} = V_{Pr,N_{V_{Pr}}}$, тобто $\forall$ j : $F_{Pr}(m_{\mathfrak{B}_2,centr,v_l,j}, current) = F_{Pr}(m_{\mathfrak{B}_2,centr,v_l,j}, N_{V_{Pr}}))$ *and* $(\{p_{Pr,1}\} \cup \{p_{Pr,2}\} \cup \ldots \cup \{p_{Pr,N_{P_{Pr}}}\})$ and $(\{A_{1,1}^{\mathfrak{S}}, A_{1,2}^{\mathfrak{S}}, \ldots, A_{1,N_{1,A^{\mathfrak{S}}}}^{\mathfrak{S}}\} \cup \{A_{2,1}^{\mathfrak{S}}, A_{2,2}^{\mathfrak{S}}, \ldots, A_{2,N_{2,A^{\mathfrak{S}}}}^{\mathfrak{S}}\})$ and $(0 \leq F_{kr}^{centr} \leq 1)$ *and* $(F_{OD}^{centr}(V_{Pr,next2}, M_{V_{Pr}}, P'_{Pr}) \geq F_{OD}^{centr}(V_{Pr,next1}, M_{V_{Pr}}, P'_{Pr}))$ *and* $(0 \leq F_{OD}^{centr}(V_{Pr,next}, M_{V_{Pr}}, P'_{Pr}) \leq 1)$ *and* $((F_{OD,t}^{centr}(V_{Pr,next}) = 0)$ or $\frac{t-(F_{OD,t}^{centr}(V_{Pr,next})}{t} \leq e_1^{centr})$ and $((F_{OD,k}^{centr} = 0)$ or $(\frac{F_{OD,k}^{centr}}{F_{OD,z}^{centr}} \leq e_1^{centr}))$ and $(N_A^{centr} \geq 2)$ and $(F_P^{centr}(V_{Pr,current}, V_{Pr,next}, M_{V_{Pr}}, P_{Pr}, A^{\mathfrak{S}}) = 1)$ *and* $(\frac{N_A^{centr}}{N} \geq e_1^{centr})$ and $((n_{s,current}^{centr} = n_{s,next}^{centr})$ *or* $((n_{s,current}^{centr} \neq n_{s',next}^{centr}$ and s = 1, …,4 and $s' = 1, \ldots, 4$ *and* $s' \neq s)$ *and* $n_{s',next}^{centr}))$ *and* $(\frac{k_m}{N_{M_{str,next}^{centr}} \sum_{m=1} k_m} \leq e_2^{centr}) \Rightarrow V_{Pr,next}$;

$m_{Pr,5} = \forall A^{\mathfrak{S}}$: $((g_1^{centr} = \text{false})$ *and* $(g_2^{centr} = \text{true})$ *and* $(g_3^{centr} = \text{false}))$ $\Rightarrow m_{Pr,s}$ *and* $V_{Pr,nast'} \neq V_{Pr,next}$: $\exists$ j $F_{Pr}(m_{\mathfrak{B}_2,centr,v_l,j}, nast') \neq F_{Pr}(m_{\mathfrak{B}_2,centr,v_l,j}, next)) \Rightarrow V_{Pr,next}$;

$m_{Pr,6} = \forall A^{\mathfrak{S}}$: $((g_1^{centr} = \text{false})$ *and* $(g_2^{centr} = \text{false})$ *and* $(g_3^{centr} = \text{true}))$ *and* $(e_3^{centr} > 2)$ and *mod* $(m_{Pr,1}$ *or* $m_{Pr,2}$ *or* $m_{Pr,3}$ *or* $m_{Pr,4}$ *or* $m_{Pr,5})$, (s = 1, …,4),

where g_l^{centr} – group of rules (l=1,2,3); $m_{Pr,i}$ – i-th rule for choosing a centralization option in multicomputer systems; i $i = 1,2, \ldots, N_{M_{Pr}}$; $N_{M_{Pr}}$ – number of rules in the set of rules M_{Pr}; where l – number of sets (l=1,2,…10);

j_l-th element in the set $M_{\mathfrak{B}_2,centr,l}$; i – system state number; $V_{Pr,current}$ – vector that specifies the state of the system at the current time and is present in the state matrix $M_{V_{Pr}}$; $M_{V_{Pr}}$ – matrix of system states in the part of organizing centralization;

M_{Pr} – set of rules; $V_{Pr,next}$ – vector that specifies the next state of the system at the current time; P_{Pr} – a set of indicators that characterize the current state of the system and processes in it; $p_{Pr,i}$ – the i-th indicator ($i = 1,2,\dots,N_{P_{Pr}}$), which characterizes the current state of the system and processes in it; $N_{P_{Pr}}$ – the number of indicators that characterize the current state of the system and processes in it and affect the change in the centralization option in the system; $A_{1,i}^{\mathfrak{S}}$ – the i-th component ($i = 1,2,\dots,N_{1,A^{\mathfrak{S}}}$) of the subset $A_1^{\mathfrak{S}}$ of the $A^{\mathfrak{S}}$ system of class $\mathfrak{S}$; $N_{1,A^{\mathfrak{S}}}$ – the number of elements in the subset $A_1^{\mathfrak{S}}$; $A_{2,j}^{\mathfrak{S}}$ – the j-th component ($j = 1,2,\dots,N_{2,A^{\mathfrak{S}}}$) of the subset $A_2^{\mathfrak{S}}$ of the $A^{\mathfrak{S}}$ system of class $\mathfrak{S}$; $N_{2,A^{\mathfrak{S}}}$ – the number of elements in the subset $A_2^{\mathfrak{S}}$; $N_{A^{\mathfrak{S}}} = N_{1,A^{\mathfrak{S}}} + N_{2,A^{\mathfrak{S}}}$; $next_1$, $next_2$ – the numbers of the last identical centralization options in the system architecture, which can be in the next option again; s'=1,…,4; $m_{str,nast,m}^{centr}$ – the m-th element of the set of strategies $M_{str,next}^{centr}$; $m = 1,2,\dots,N_{M_{str,nast}^{centr}}$; $N_{M_{str,nast}^{centr}}$ – the number of strategies for choosing the next centralization option; k_m – the number of applications of a certain m – the strategy for choosing a centralization option; $V_{Pr,next'}$) – the vector of the next centralization option to which the system could not proceed, i.e. the option that did not take place; *mod* – function for changing elements in the rule argument; e_1^{centr}=0.25 – first threshold value; e_2^{centr}=0.1 – second threshold value; e_3^{centr}=2 – third threshold value.

E. METHOD FOR EVALUATING THE NEXT CENTRALIZATION

The obtained rules allow class $\mathfrak{S}$ systems to independently determine the next centralization option. At the same time, the rules take into account not only the successful transition of the system to the next centralization option, but also the option when the system was unable to make the transition to the next centralization option. Also, the system can choose a rule from the third group of rules when it is influenced for a long time and the previous versions of the rules cannot ensure the formation of its center in such a way as to stabilize its functioning. The modification of the basic rules is carried out by partially or completely changing their parameters. The developed rules take into account the specifics of class $\mathfrak{S}$ systems, which are distributed and designed to function independently under malicious influences without administrator intervention.

According to the obtained rules, we will set the main steps of the method for determining the next centralization option in class $\mathfrak{S}$ systems as follows:

1) the occurrence of events (receiving influences or instructions) to start changing the system architecture;

2) if the system controller has approved the decision to change the centralization option in the system, then go to step 3), otherwise go to step 9);

3) determine the type of event and set the rule for the class;

4) if the event to change the centralization option in the system involves a planned transition, then apply the rules $m_{Pr,s}$ (s=1,…,4) to prepare five centralization options;

5) if the event to change the centralization option in the system is related to the fact that the transition to the next centralization option in the system did not occur or cannot be completed, then apply the rule $m_{Pr,5}$ to prepare five centralization options;

6) if the event regarding the change of the centralization option in the system is related to the fact that the transition to the next centralization option in the system did not occur or cannot be completed for a long time and multiple application (exceeding the threshold value) of the rule $m_{Pr,5}$ did not yield results, then apply the rule $m_{Pr,6}$ to prepare five centralization options;

7) if one of the steps 4)-6) is successfully completed, then proceed to step 8), otherwise remain in the current state and start executing step 1);

8) transfer of 5 potential centralization options in the system to the controller for approval of the next centralization option;

9) proceed to determining the option for changing the rest of the architecture in the system that does not contain the center of system.

The essence of the method for determining the next centralization option in class $\mathfrak{S}$ systems is to use rules. The rules ensure the avoidance of partial or complete enumeration of options from possible centralization options. To achieve the result, complex criteria of efficiency, stability, integrity, and security were used. Also, the division of the architecture type into centralized, partially centralized, partially decentralized, and decentralized was taken into account. This allows, according to the rules for selecting the centralization option, to evaluate each of the selected options depending on the number of active system components at the current time and the criteria. As a result, it becomes possible to select the next centralization option from a large number of options without evaluating all options. This ensures speed and avoidance of complete or significant partial enumeration of all options in a constantly changing environment.

IV. EXPERIMENTS

A. EXPERIMENT SETUP

For the developed method for determining the centralization option in class $\mathfrak{S}$ systems, we will carry out an experiment. The objectives of the experiment will be as follows: establishing the next centralization option during the specified time of system operation; time spent on determining the next centralization option and switching to it; number of successful attempts to switch to the next centralization option; number of unsuccessful attempts to switch to the next centralization option; values of the objective function and indicators from the rules; rule numbers used to determine the next centralization options. The obtained results of the experimental tasks will allow us to assess the possibility of implementing the developed method in class $\mathfrak{S}$ systems and taking into account previous experience in choosing a centralization option.

First, let's establish a list of indicators that will be used when applying the method for determining the next option for centralization in the system.

B. RESULTS

To conduct the experiment, a multicomputer system was implemented with functionality that allows accepting influences from outside and inside the corporate network. The system architecture separates the center of system and the decision-making controller. The center of system prepares five options for performing the task, which is a response to an event caused by influences. The decision-making controller approves one option for performing the task from the five proposed options. The multicomputer system operated in the corporate

network for 90 days. In Table. 1 presents the results of its operation regarding:

1) the total number of options for restructuring the center of system during the operation time (column 1);
2) the start time of restructuring the center of system (column 2);
3) five options for the next option of centralization in the system architecture (numbering of 5 rows, column 3);
4) ten characteristic properties from formula (1) for each option of centralization (columns 4-13);
5) the number of the approved option of centralization in the system architecture (column 14);
6) the completion time of restructuring the center of system (column 15);
7) the execution time of restructuring the center of system (column 16);
8) the result regarding the successful implementation of restructuring the center of system (successful attempt - "1", unsuccessful attempt - "0", column 17);
9) the number of the option of the rule used when choosing the next option of centralization in the system architecture (column 18);
10) the objective function F_{kr}^{centr} for evaluating the next options of centralization for choosing 1 of the options from 4 types of centralization (formula (12), column 19);
11) number of the centralization option with which the same option was re-approved (column 20);
12) number of the rule option ("1" - 1..4, "2" - 5, "3" - 6) in three groups (column 21).

Let us summarize the results of the multicomputer system operation in Table 2. It contains the following data: the total number of center of system reorganizations during its operation (column 1); the number of options used for each characteristic property (columns 4-13); information on the success/failure of the center of system reorganization (column 17); the value of the objective function (the smallest, approved, largest among the five options, column 19); the number of centralization options by numbers with which the same option was re-approved (column 20); the number of rule option numbers ("1" - 1..4, "2" - 5, "3" - 6) by three groups (column 21).

The experiment conducted confirms the stability of the system. In particular, the system was unable to complete the reconstruction of its center 35 times out of 330 times. But in 34 cases, completion occurred from the second updated attempt, and in one case a third attempt was required, the execution of which was carried out according to the sixth rule. Also, the values of the objective function for all five options that were prepared by the center of system were in the interval 0-0.25. For the case of successful reconstruction of the center of system, the upper limit of the interval did not exceed 0.1, and for the case of repeated selection of reconstruction options due to problems with completing the reconstruction of the center of system - 0.25. These values in the context of the objective function confirm the adequacy of the selection of options for selection by the decision-making controller and use for the next centralization option in the system architecture. The graph of the dependence of the reconstruction time and the values of the objective function is shown in Fig. 2.

According to the results of the second experiment, it was found that the values of the objective function were mainly in the range of 0-0.72, which are quite large and indicate a certain imbalance in the choice of the centralization option. A comparison of the two experiments is shown in Fig. 4 and Fig. 5. The graphs for the approved centralization options differ in that in the first experiment the value of the objective function may not necessarily be chosen to be the smallest, and in the second experiment - only the smallest value. This will affect the subsequent steps of the system in choosing centralization options in the system architecture and its functioning. Also, this will affect the formation of previous experience in using various parameters to choose the next centralization option.
The graph in Fig. 2 shows five values of the objective function against a certain time. The graph is discrete. The continuous line connects the values of the options approved by the system's decision-making controller.

A second experiment was conducted with the system under the same requirements as the first experiment. But in the second experiment, there was no decision-making controller in the system. The center of system prepared centralization options, calculated the values of the objective function for them, and chose the one in which the value of the objective function was minimal as the next centralization option. The results of the experiment are shown in Fig. 3.

C. RESULT ANALYSIS

Thus, for the first experiment, the values of the objective function compared to the results of the second experiment reflect better stability and stay within the interval 0-0.25 throughout the entire operation time, regardless of the fact that the next centralization options are not necessarily selected with the smallest values of the objective function. In the second experiment, the selection of the following centralization options is carried out with the selection of only the minimum values of the objective function. The results of the operation are characterized by insufficient stability of the system and an increase in the value of the objective function compared to the results of the first experiment by 0.5. This is a sufficiently large deviation.

The number of unsuccessful attempts when switching to the next option of centralization in the first case is eleven, and in the second - forty-six. This confirms the better choice in the first experiment and the advantage in using the decision controller. The chosen options for centralization in the systems architecture in the first case are more adaptable in the context of the ability to transition to them. The percentage of unsuccessful attempts compared to all attempts in the first option is 3.2%, in the second case - 13.6%. That is, due to the use of the decision-making controller to determine the next option of centralization in the systems architecture, a 10% increase in efficiency was achieved in terms of the transition of the system to the next identified option of centralization.

The time spent on the transition to the next option of centralization in the system architecture for both experiments is approximately the same. But in the second experiment, due to the large number of unsuccessful attempts to move to the next option of centralization, the time consumption is greater, since time is spent on unsuccessful attempts, the execution of which with changed options must be repeated again in the future. At the same time, additional time was spent on their implementation, which was approximately 10.5% of the total time spent on all attempts to change the option of centralization in the system architecture.

When choosing the next option of centralization in the system architecture in the first experiment, the fifth rule was applied ten times, the sixth rule - once. This means that when retrying to move to the next option of centralization in the system architecture, there were two failed attempts and therefore the sixth rule was applied. For the second experiment, the fifth rule was applied forty-five times, the sixth rule - once. This is analogous to the first experiment in the context of two consecutive failed attempts at transition. In addition, applying the fifth rule forty-five times confirms the analysis of the time spent on the failed attempts.

Table 1. Fragment of the results of the functioning of a multi-computer system (experiment 1)

Step	Start time	Variant	1-4	1-2	1-2	1-4	1-5	1-3	1-2	1-3	1-3	1-3	Number of the approved option	Completion time	Lead time	Successful attempt – 1, Unsuccessful – 0	Center variant number on approved	Meaning of the target function	Center option number, with which repetition	Rule numbers used for center options	Group number	Number of computers
1		1	1	1	2	2	4	1	1	1	1	1						0,0336				
		2	2	1	1	4	4	1	1	1	2	1						0,0268				
	29054	3	3	2	2	3	3	1	1	1	1	3	3	29059	5	1	3	0,0366		1	1	66
		4	4	2	1	4	5	1	2	3	3	2						0,0725				
		5	1	1	2	1	2	3	1	2	1	2						0,0393				
2		1	1	1	1	2	2	1	1	3	2	1						0,0922				
		2	2	1	1	2	3	1	1	2	3	2						0,0181				
		3	3	1	1	2	1	2	1	3	3	2						0,0674				
	54775	4	4	2	2	1	1	3	2	1	1	2	4	54779	4	1	4	0,0645		1	1	
		5	1	2	1	3	3	3	2	2	3	3						0,0893				
3		1	1	1	1	4	1	2	2	1	1	3						0,0507				
	80055	2	2	1	1	4	4	1	1	1	1	3	2	80060	5	1	2	0,0179		1	1	68
		3	3	1	2	1	5	3	2	2	3	2						0,0876				
		4	4	2	1	1	2	2	2	1	1	1						0,0863				
		5	3	1	2	3	4	3	2	1	3	3						0,0179				
4		1	1	2	2	1	5	2	1	1	1	1						0,0159				
		2	2	1	2	3	1	3	1	2	1	2						0,0349				
	103211	3	3	2	1	1	4	2	2	1	3	2	3	103214	3	1	3	0,0579		1	1	55
		4	4	2	2	2	1	2	2	1	1	1						0,0827				
		5	1	2	2	3	3	3	2	1	2	2						0,0569				
5		1	1	2	1	2	3	1	2	1	3	3						0,0964				
	127542	2	2	2	1	2	4	1	2	2	3	1	2	127547	5	1	2	0,0945		1	1	31
		3	3	1	1	2	1	1	1	1	1	1						0,0178				
		4	4	2	1	3	4	1	2	1	1	1						0,0856				
		5	4	2	2	1	5	3	2	2	1	2						0,0306				

Table 2. Fragment of results for functioning a multi-computer system (experiment 2)

Step	Start time	Variant	1-4	1-2	1-2	1-4	1-5	1-3	1-2	1-3	1-3	1-3	Number of the approved option	Completion time	Lead time	Successful attempt – 1, Unsuccessful – 0	Center variant number on approved	Meaning of the target function	Center option number, with which repetition	Rule numbers used for center options	Group number	Number of computers
1		1	1	1	1	1	5	3	2	2	2	1						0,5084				
	27619	2	2	1	2	3	3	2	1	2	3	3	2	27623	4	1	2	0,4194		1	1	39
		3	3	2	1	3	5	3	1	1	2	3						0,4601				
		4	4	1	1	2	4	3	2	2	1	1						0,5579				
		5	4	2	2	2	5	1	2	3	2	2						0,5888				
2		1	1	2	1	1	3	3	2	2	1	1						0,5845				
		2	2	2	2	4	2	3	2	3	2	1						0,3045				
		3	3	2	2	4	2	2	1	2	1	1						0,4785				
		4	4	2	1	1	5	2	1	3	1	2						0,348				
	56213	5	3	2	2	3	5	3	1	1	2	1	5	56218	5	1	3	2,78E-04		1	1	108
3		1	1	1	2	2	4	3	1	2	2	2						0,5299				
		2	2	2	1	2	2	2	1	3	3	2						0,19				
	81992	3	3	1	2	2	1	3	1	2	3	1	3	81995	3	1	3	0,0426		1	1	98
		4	4	1	2	4	2	3	2	3	3	1						0,6311				
		5	3	2	1	2	1	3	2	1	2	3						0,4833				
4		1	1	2	2	3	5	3	1	2	3	3						0,2666				
		2	2	2	2	2	1	1	1	2	3	1						0,08				
	106505	3	3	2	1	3	2	3	1	1	3	3	3	106510	5	1	3	0,0644		1	1	22
		4	4	1	1	3	1	2	1	3	1	2						0,4541				
		5	2	2	1	3	2	2	1	3	1	3						0,7091				
5		1	1	2	1	1	5	3	2	2	1	3						0,4118				
		2	2	1	2	1	5	1	2	1	1	3						0,61				
	136331	3	3	2	1	2	2	3	2	2	2	3	3	136334	3	0	3	0,198		1	1	105
		4	4	1	1	1	3	1	1	1	2	2						0,4333				
		5	1	1	2	3	2	1	1	3	2	2						0,3701				

Table 3. Summary of experiment 1

Step	Start time	Variant	1-4	1-2	1-2	1-4	1-5	1-3	1-2	1-3	1-3	1-3	Number of the	Completion time	Lead time	Successful attempt – 1, Unsuccessful – 0	Center variant	Meaning of the target function	Center option number, with which repetition	Rule numbers used for center options	Group number	Number of computers
337			**1:**	**1:**	**1:**	**1:**	**1:**	**1:**	**1:**	**1:**	**1:**	**1:**			**SUM:**	**1:**	**1:**	**Aver:**		**1:**	**1:**	
			388	756	740	381	283	517	764	505	510	511			1404	296	70	0,0547		245	296	
			2:	**2:**	**2:**	**2:**	**2:**	**2:**	**2:**	**2:**	**2:**	**2:**				0:	2:	**Aver_f_zatv:**		**2:**	**2:**	
			381	779	795	390	315	497	771	517	522	519				11	74	0.0530		16	10	
			3:			**3:**	**3:**	**3:**		**3:**	**3:**	**3:**					3:			**3:**	**3:**	
			384			393	322	521		513	503	505					82			17	1	
			4:			4:	**4:**										4:			**4:**		
			382			371	317										81			18		
							5:													**5:**		
							298													10		
																				6:		
																				1		

Table 4. Summary of experiment 2

Step	Start time	Variant	1-4	1-2	1-2	1-4	1-5	1-3	1-2	1-3	1-3	1-3	Number of the	Completion time	Lead time	Successful attempt – 1, Unsuccessful – 0	Center variant	Meaning of the target function	Center option number, with which repetition	Rule numbers used for center options	Group number	Number of computers
337			**1:**	**1:**	**1:**	**1:**	**1:**	**1:**	**1:**	**1:**	**1:**	**1:**			SUM:	**1:**		**Aver:**		**1:**	**1:**	
			422	817	853	398	335	549	839	540	570	569			2062	291		0,3553		236	291	
			2:	**2:**	**2:**	**2:**	**2:**	**2:**	**2:**	**2:**	**2:**	**2:**				**0:**				**2:**	**2:**	
			414	868	832	438	344	550	846	579	565	571				46				17	45	
			3:			**3:**	**3:**	**3:**		**3:**	**3:**	**3:**						**Aver_f_min:**		**3:**	**3:**	
			432			411	359	586		566	550	545						0,1149		16	1	
			4:			**4:**	**4:**													**4:**		
			417			438	331													22		
							5:													**5:**		
							316													45		
																				6:		
																				1		

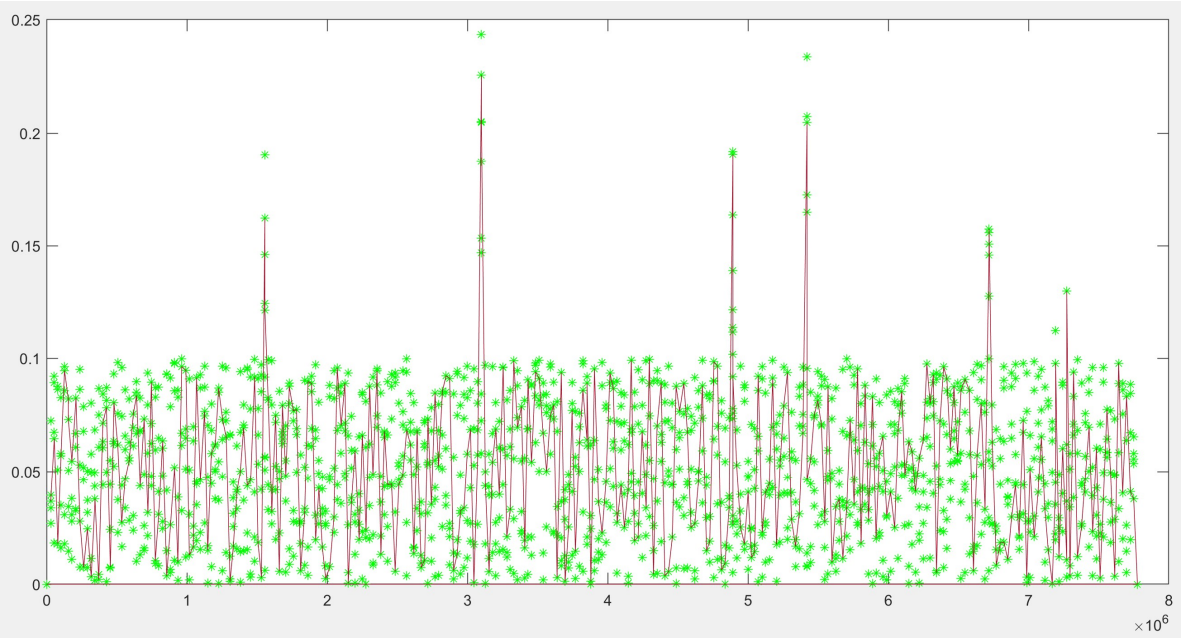


Figure 2. Results of the Experiment 1.

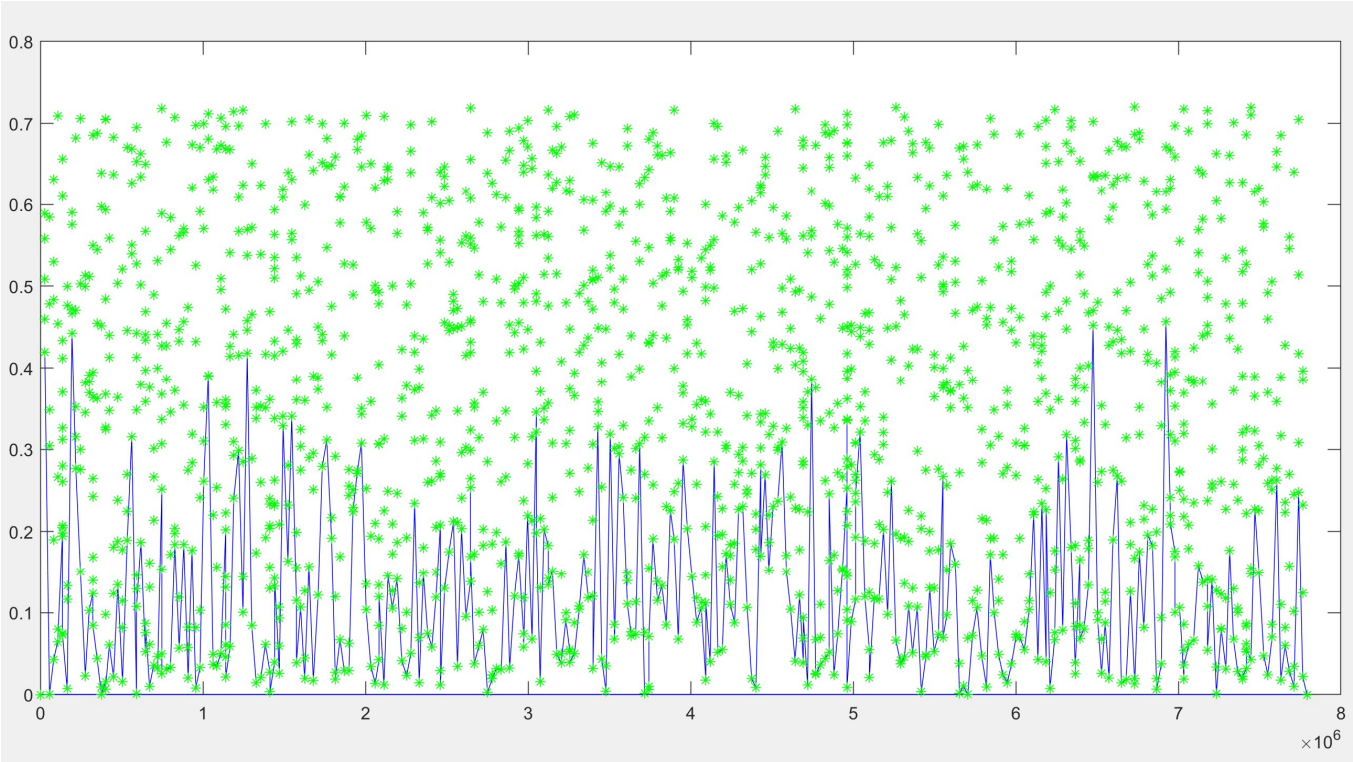


Figure 3. Results of the Experiment 2.

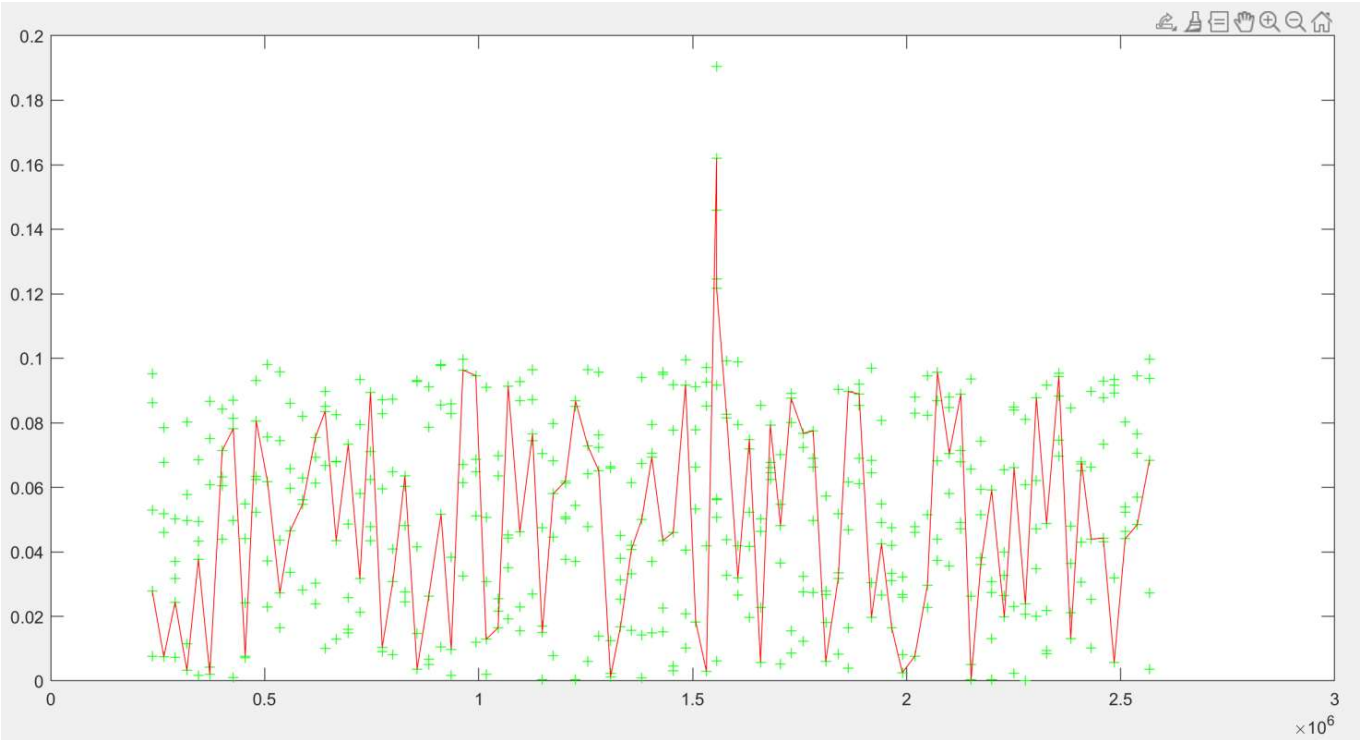


Figure 4. Fragment of objective function graph for the first experiment (10-100 steps).

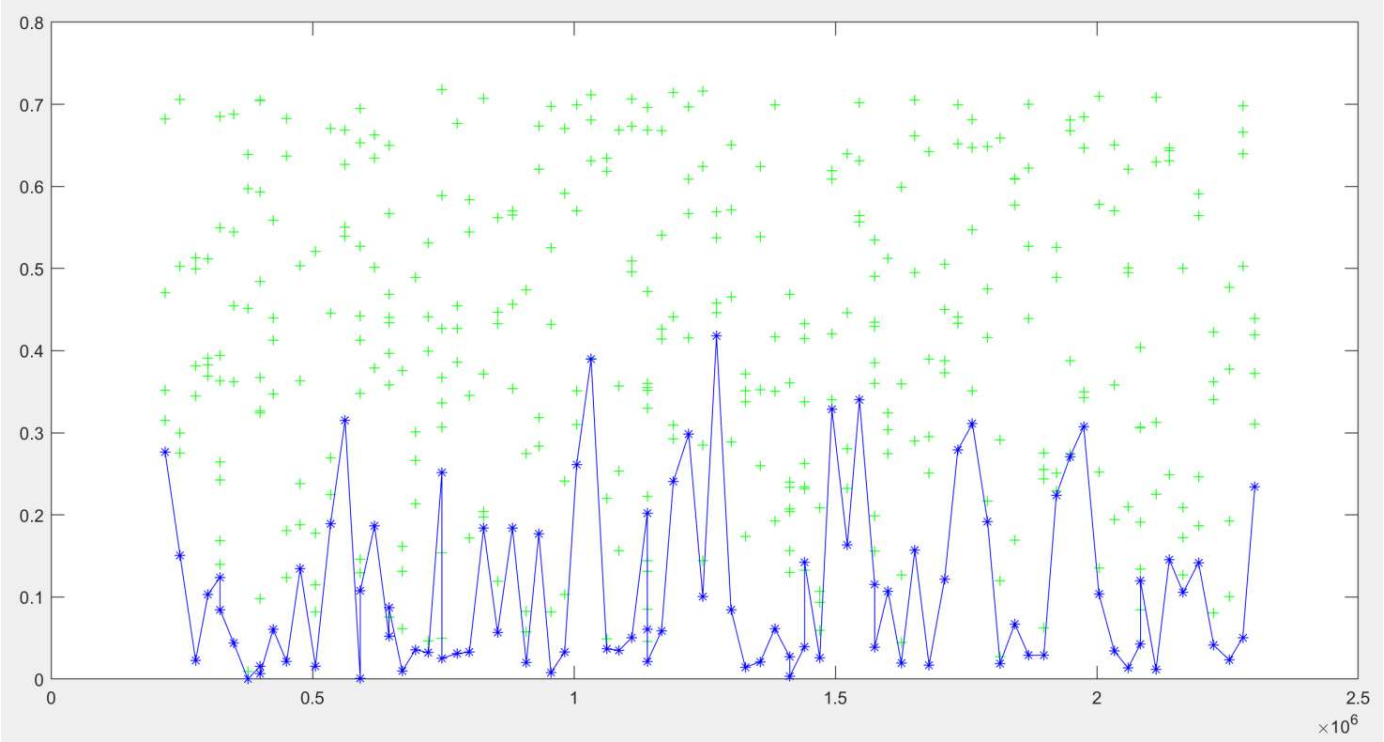


Figure 5. Fragment of objective function graph for the second experiment (10-100 steps).

V. DISCUSSION

The choice of the next centralization option in the architecture of multi-computer systems using a decision-making controller ensures further stable operation of the system. This is reflected in the values of the objective function, the variance of which is in a small interval. The results of the experiment confirm the correctness and adequacy of using the developed objective function to evaluate the next centralization option in the system architecture. The difference between the variances in both experiments is approximately 50%. This makes it possible to unambiguously single out the use of a decision-making controller as an appropriate component of systems when choosing the next centralization option. In further research, it is advisable to improve the complex criteria of efficiency, stability, integrity, and security that form the objective function to reduce the variance.

VI. CONCLUSIONS

A method of determining the option of centralization in multi-computer systems of antivirus combined baits and traps has been developed. The peculiarity of the method is that the choice of the next option of centralization is carried out according to complex criteria of efficiency, stability, integrity, and security. At the same time, the division of the type of system architecture into centralized, partially centralized, partially decentralized, and decentralized was also taken into account. This makes it possible to evaluate each of the selected options depending on the number of active components of the systems at the current time and criteria according to the developed rule for choosing the option of centralization. As a result, it was possible to select the next option of centralization from a large number of options. This selection of the next option of centralization became possible without evaluating all options. This ensures speed and avoids a complete search of all options in a dynamically changing environment.

The results of the experiment when using the decision-making controller as part of the system to select the next option of centralization when rebuilding the system architecture confirm the effectiveness of the developed rule and method of choosing the next option of centralization. In particular, in the case of using a decision-making controller, the value of the objective function for all considered options is 50% less compared to the option without a controller. Due to the use of the decision-making controller, the selection of successful options for reconstruction and the time for their implementation was achieved by approximately 10%.

VII. FUTUREWORK

Further research will concern the development of the controller's architecture and its decision-making method. This will make it possible for the system to analyze the variants of tasks received from the center, taking into account previous experience, and to approve one of the solutions for implementation. Among such options for performing tasks, there is also an option to choose centralization. For it, it is necessary to improve the parameters of the complex criteria of efficiency, stability, integrity, security, which form the target function, in order to reduce dispersion.

References

[1] A. Kashtalian, S. Lysenko, O. Savenko, A. Nicheporuk, T. Sochor, and V. Avsiyevych, "Multi-computer malware detection systems with metamorphic functionality," *Radioelectronic and Computer Systems*, vol. 2024, no. 1, pp. 152–175, 2024. https://doi.org/10.32620/reks.2024.1.13.

[2] O. Savenko, A. Sachenko, S. Lysenko, G. Markowsky, and N. Vasylkiv, "Botnet detection approach based on the distributed systems," *International Journal of Computing*, vol. 19, no. 2, pp. 190–198, 2020. https://doi.org/10.47839/ijc.19.2.1761.

[3] B. Savenko, A. Kashtalian, S. Lysenko, and O. Savenko, "Malware detection by distributed systems with partial centralization," in *2023 IEEE 12th International Conference on Intelligent Data Acquisition and*

Advanced Computing Systems: Technology and Applications (IDAACS), Dortmund, Germany, 2023, pp. 265–270. https://doi.org/10.1109/IDAACS58523.2023.10348773.

[4] A. Kashtalian, S. Lysenko, B. Savenko, T. Sochor, and T. Kysil, "Principle and method of deception systems synthesizing for malware and computer attacks detection," *Radioelectronic and Computer Systems*, vol. 2023, no. 4, pp. 112–151, 2023, doi: https://doi.org/10.32620/reks.2023.4.10.

[5] N. Doukas, P. Stavroulakis, & N. Bardis, "Review of artificial intelligence cyber threat assessment techniques for increased system survivability," *Malware Analysis Using Artificial Intelligence and Deep Learning*, Springer International Publishing, 2021, pp. 207-222. https://doi.org/10.1007/978-3-030-62582-5_7.

[6] R. Mehresh and S.J. Upadhyaya, "Deception-based survivability," in *Secure System Design and Trustable Computing*, C.H. Chang and M. Potkonjak, Eds. Cham: Springer, 2016, https://doi.org/10.1007/978-3-319-14971-4_17.

[7] N. Soule, P. Pal, S. Clark, B. Krisler, and A. Macera, "Enabling defensive deception in distributed system environments," in *Proceedings of the 2016 Resilience Week (RWS)*, Chicago, IL, USA, 2016, pp. 73–76, https://doi.org/10.1109/RWEEK.2016.7573310.

[8] M. Baykara and R. Das, "SoftSwitch: A centralized honeypot-based security approach using software-defined switching for secure management of VLAN networks," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 27, pp. 3309–3325, 2019, https://doi.org/10.3906/elk-1812-86.

[9] S. Kyung, W. Han, N. K. Tiwari, V. H. Dixit, L. Srinivas, Z. Zhao, A. Doupé, and G. Ahn, "HoneyProxy: Design and implementation of next-generation honeynet via SDN," in *Proceedings of the 2017 IEEE Conference on Communications and Network Security (CNS)*, 2017, pp. 1–9. https://doi.org/10.1109/CNS.2017.8228653.

[10] N. H. Khoa, H. Do, K. Ngo-Khanh, P. T. Duy, and V.H. Pham, "SDN-based cyber deception deployment for proactive defense strategy using Honey of Things and cyber threat intelligence," in *Intelligence of Things: Technologies and Applications, ICIT 2023*, vol. 188, N.N. Dao, T.N. Thinh, and N.T. Nguyen, Eds. Cham: Springer, 2023, https://doi.org/10.1007/978-3-031-46749-3_26.

[11] C. Gao, Y. Wang, X. Xiong, and W. Zhao, "MTDCD: An MTD enhanced cyber deception defense system," in *Proceedings of the 2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Chongqing, China, 2021, pp. 1412–1417, https://doi.org/10.1109/IMCEC51613.2021.9482133.

[12] F. De Gaspari, S. Jajodia, L. Mancini, and A. Panico, "AHEAD: A new architecture for active defense," in *Proceedings of the ACM International Conference on Computer and Communications Security*, 2016, pp. 11–16, https://doi.org/10.1145/2994475.2994481.

[13] W. Fan, D. Fernández, and Z. Du, "Adaptive and flexible virtual honeynet," in *Advances in Computer Science and Ubiquitous Computing*, Cham: Springer, 2015, https://doi.org/10.1007/978-3-319-25744-0_1.

[14] D. Sever and T. Kišasondi, "Efficiency and security of docker-based honeypot systems," in *Proceedings of the 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, 2018, pp. 1167–1173, https://doi.org/10.23919/MIPRO.2018.8400212.

[15] A. Ravi, B. Sharma, and A. Mukherjee, "A cloud-native honeynet automation and orchestration framework," *OSF Preprints*, 2023, https://doi.org/10.31219/osf.io/xkqzr.

[16] T. Yu, Y. Xin, and C. Zhang, "HoneyFactory: Container-based comprehensive cyber deception honeynet architecture," *Electronics*, vol. 13, no. 2, p. 361, 2024, https://doi.org/10.3390/electronics13020361.

[17] M.M. Islam and E. Al-Shaer, "Active deception framework: An extensible development environment for adaptive cyber deception," in *Proceedings of the 2020 IEEE Secure Development (SecDev)*, Atlanta, GA, USA, 2020, pp. 41–48, https://doi.org/10.1109/SecDev45635.2020.00023.

[18] J. Landsborough, L. Carpenter, B. Coronado, S. Fugate, K. Ferguson-Walter, and D. Bruggen, "Towards self-adaptive cyber deception for defense," in *Proceedings of the HICSS-54*, 2021, https://doi.org/10.24251/HICSS.2021.244.

[19] C.J. Chiang, Y.M. Gottlieb, S. Sugrim, R. Chadha, C. Serban, A. Poylisher, L.M. Marvel, and J. Santos, "ACyDS: An adaptive cyber deception system," in *Proceedings of the 2016 IEEE Military Communications Conference MILCOM 2016*, 2016, pp. 800–805, https://doi.org/10.1109/MILCOM.2016.7795427.

[20] W. Tounsi, "Cyber deception: The ultimate piece of a defensive strategy – proof of concept," in *Proceedings of the 2022 7th International Conference on Cyber Security and Protection of Digital Services (CyberSec)*, 2022, pp. 1–5, https://doi.org/10.1109/CSNet56116.2022.9955605.

[21] A. Mitropoulos, G. Kougka, D. Lampoudis, and C. Patsakis, "Design and implementation of adaptive deception-based cyber security solutions," in *Proceedings of the 2021 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, Dubai, UAE, 2021, pp. 247–252. doi: 10.1109/ICIoT52221.2021.9476067.

[22] S. Nazemi, A. Ghafoorian, and A. Azmoodeh, "A dynamic cyber deception method for APT defense," in *Proceedings of the 2020 6th International Conference on Web Research (ICWR)*, Tehran, Iran, 2020, pp. 128–133, https://doi.org/10.1109/ICWR49608.2020.9122318.

[23] Z. Sui, X. Wang, Y. Zhang, and Z. Zhang, "Dynamic cyber deception based on knowledge graph reasoning," in *Proceedings of the 2021 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 2021, pp. 1–6, doi: 10.1109/ISI54217.2021.9674327.

[24] J. Haslum, R. Dewar, S. Chaudhry, and I. Ray, "Exploring active cyber deception for security: A survey of tools and techniques," *IEEE Access*, vol. 8, pp. 22444–22458, 2020, doi: 10.1109/ACCESS.2020.2969643.

[25] L. Chen, J. Xu, and Y. Liu, "A survey of deception-based defense mechanisms," *Security and Communication Networks*, vol. 2021, Article ID 5562934, 2021, doi: 10.1155/2021/5562934.

[26] T. Zheng and H. Liu, "Game-based approach for active cyber defense using deception techniques," in *Proceedings of the 2020 IEEE Symposium on Privacy-Aware Computing (PAC)*, 2020, pp. 25–31, doi: 10.1109/PAC50185.2020.00010.

[27] M. Manadhata, J. Kim, A. Mettler, and D. Melski, "Deceptive defense design," *Journal of Cybersecurity and Privacy*, vol. 1, no. 1, pp. 66–85, 2021, https://doi.org/10.3390/jcp1010005.

[28] S. Rajagopalan, S. Yu, and H. Zheng, "Dynamic deception for proactive defense," in *Proceedings of the 2017 IEEE International Symposium on Cyberspace Safety and Security (CSS)*, 2017, pp. 41–47, doi: 10.1109/CSS.2017.19.

[29] M. Prandini and M. Drahansky, "The role of deception in computer security: A review," *Computers & Security*, vol. 109, 102396, 2021, https://doi.org/10.1016/j.cose.2021.102396.

[30] H. Zheng and L. Liu, "Moving target defense with deception technology," in *Proceedings of the 2021 ACM Workshop on Moving Target Defense (MTD)*, 2021, pp. 1–6, doi: 10.1145/3485947.3485954.

[31] P. Frazier, K. Ferguson-Walter, and J. Landsborough, "The effectiveness of adaptive cyber deception tactics," in *Proceedings of the 2022 Annual Computer Security Applications Conference (ACSAC)*, 2022, pp. 203–214, doi: 10.1145/3564625.3564677.

[32] S. Hossain and J. Lee, "Application of machine learning techniques in deception-based cybersecurity," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 2, pp. 382–392, 2022, doi: 10.1109/TETCI.2022.3148418.

[33] W. Zhao, Y. Xu, and H. Yu, "Integrating deception techniques into multi-layered security frameworks," in *Proceedings of the 2023 IEEE International Conference on Cloud Computing (CLOUD)*, 2023, pp. 95–100, doi: 10.1109/CLOUD57382.2023.10151620.

[34] Y. Wang, G. Yang, J. Chen, and L. Liu, "The role of deception in cybersecurity defense: A critical review," *Future Generation Computer Systems*, vol. 142, pp. 234–247, 2024, https://doi.org/10.1016/j.future.2023.12.018.

[35] A. Sachenko, V. Kochan, V. Kharchenko, H. Roth, V. Yatskiv, M. Chernyshov, P. Bykovyy, O. Roshchupkin, V. Koval, H. Fesenko, "Mobile post-emergency monitoring system for nuclear power plants," in *CEUR Workshop Proceedings*, vol. 1614, pp. 384–398, 2016. CEUR-WS. ISSN: 1613-0073.

[36] L. Huang and Q. Zhu, "Duplicity games for deception design with an application to insider threat mitigation," in *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4843-4856, 2021, https://doi.org/10.1109/TIFS.2021.3118886.

[37] P.S. Park, S. Goldstein, A. O'Gara, M. Chen, D. Hendrycks, "AI Deception: A Survey of Examples, Risks, and Potential Solutions," https://doi.org/10.48550/arXiv.2308.14752.

[38] L. Zhang, V. L. L. Thing, "Three Decades of Deception Techniques in Active Cyber Defense – Retrospect and Outlook," https://doi.org/10.48550/arXiv.2104.03594.

[39] B. C. Ward, S. R. Gomez, R. W. Skowyra, D. Bigelow, J. Martin, J. Landry, H. Okhravi, "Survey of Cyber Moving Targets Second Edition," Semantic Scholar, 2018, https://www.semanticscholar.org/paper/Survey-of-Cyber-Moving-Targets-Second-Edition-Ward-Gomez/889aa42d8b9e7f731324cca810071333544c4f60.

[40] P. V. Mohan, S. Dixit, A. Gyaneshwar, U. Chadha, K. Srinivasan, J. T. Seo, "Leveraging computational intelligence techniques for defensive deception: A review, recent advances, open problems and future

directions," *Sensors*, vol. 22, no. 6, 2194, 2022, https://doi.org/10.3390/s22062194.
[41] D. Jay, "Deception technology based intrusion protection and detection mechanism for digital substations: A game theoretical approach," in *IEEE Access*, vol. 11, pp. 53301-53314, 2023, https://doi.org/10.1109/ACCESS.2023.3279504.
[42] M. R Amal, P. Venkadesh, "Review of cyber attack detection: Honeypot system," *Webology*, vol. 19, no. 1, pp. 5497-5514, 2022, https://doi.org/10.14704/WEB/V19I1/WEB19370.

ANTONINA KASHTALIAN. PhD, Associate Professor at the Department of Physics and Electrical Engineering, Doctoral Staff, Khmelnytskyi National University, Khmelnytskyi, Ukraine. Her research interests cover deception systems in cybersecurity, artificial intelligence, machine and deep learning.

SERGII LYSENKO. Full Doctor, Full Professor of the Computer Engineering & Information Systems Department in Khmelnytskyi National University. The author has been working for more than 15 years in the field of increasing the effectiveness of detecting malicious software and cyberattacks, in particular: the theory and practice of creating multi-agent and distributed systems for detecting malicious software in computer networks, creating an adaptive technology for detecting cyber threats in computer networks.

TETIANA KYSIL. PhD, Associate Professor at the Department of Computer Engineering and Information Systems, Khmelnytskyi National University, Khmelnytskyi, Ukraine. His research interests encompass the computer simulation.

ANATOLIY SACHENKO. Doctor of Technical Sciences, Honored Inventor of Ukraine, Professor at the West Ukrainian National University, Director of the Research Institute of Intelligent Computer Systems, Ternopil, Ukraine. His research interests include Computational Intelligence in Applications, Cyber Security, IoT Sensors, IT Project Management.

OLEG SAVENKO. Doctor of Technical Sciences, Professor at the Department of Computer Engineering and Information Systems, Khmelnytskyi National University, Khmelnytskyi, Ukraine. His research interests encompass the development of distributed malware detection systems and the design of intelligent information systems for object detection using drones.

Bohdan Savenko PhD, Associate Professor at the Department of Computer Engineering and Information Systems, Khmelnytskyi National University, Khmelnytskyi, Ukraine. His research interests are the methods and systems for detecting malicious software in computer networks.