Implementation of combined method in constructing a trajectory for structure reconfiguration of a computer system with reconstructible structure and programmable logic

© Volodymyr Tokariev, © Vitalii Tkachov, © Iryna Ilina, © Stanislav Partyka

Kharkiv National University of Radio Electronics, Kharkiv, Ukraine
tkachov@ieee.org, tokarev@ieee.org, ilirvi_71@ukr.net,
stanislav.partyka@nure.ua

Abstract. In modern conditions providing the continuity of technological processes as well as increasing the reliability and survivability of operation of computer systems with configurable structure and programmable logic is one of the main strategic directions of modern social-economic and technical complexes. This is conditioned by the need to maintain the resilience and stability in the operation of computer systems with a reconstructible structure and programmable logic under various conditions of adverse effects of external and internal factors, both natural and technogenic, as well as those human-induced. It has been found that neutralizing threats and minimizing losses caused by abnormal, critical, emergency and catastrophic situations leading to an avalanche-like increase in degradation processes and destruction of computer systems with a reconstructible structure and programmable logic, requires the development of new principles, approaches, ways and methods of operational monitoring, analysis and forecasting of situations, development of options for control decisions, procedures for their selection and implementation in the framework of theory of structural dynamics control. To solve the optimization problem of constructing scenarios for the structural reconfiguration of computer systems with a reconstructible structure and programmable logic, a method and an algorithm that implements this method are proposed. The novelty of the method is the combined use of the random guided search method and the method of cutting -off unpromising variants of structural reconfiguration of computer systems with a reconstructible structure and programmable logic such as « branch and bound». The proposed approach allows to solve the optimization problems of constructing scenarios for the structural reconfiguration of computer systems with a reconstructible structure and programmable logic, both unconditional and conditional, that can be described using various structural and topological indicators.

Keywords: combined method, reconfiguration, functional element, structure, computer system, survivability, graph.

1 Introduction

The analysis of existing and predicted critical and emergency situations that are currently occurring everywhere in various subject areas shows that they stop being sectoral, and develop into accidents and disasters that are already intersectoral in nature. Under these conditions, it is necessary to investigate and solve the problems of increasing the reliability and survivability of computer systems with a reconstructible structure and programmable logic within the framework of the interdisciplinary approach. The causes that lead to the emergence of critical situations, accidents and disasters, which have natural-ecological, technical-industrial or anthropogenic-social causes, are of particular danger to modern computer systems with a reconstructible structure and programmable logic. Moreover, the range of threats to economic, physical and information security, as well as the list of vulnerabilities in the hardware-software and information structures of computer systems with a reconstructible structure and programmable logic is constantly growing [1-4]. Thus, in real life, situations are possible when these threats are combined and lead to an avalanche-like occurrence and development of negative events, ultimately resulting in catastrophic consequences. Under these conditions, ensuring the continuity of technological processes and increasing the reliability and survivability of the corresponding production systems is one of the most important strategic directions in the development of modern socio-economic and technical complexes. This is caused by the need to maintain the resilience and stability in the functioning of computer systems with a reconstructible structure and programmable logic under various conditions of adverse effects of external and internal factors, which can be anthropogenic and/or natural.

2 Analysis of known solutions

In practice, when solving the problems of ensuring survivability, computer systems with a reconstructible structure and programmable logic have received such an option for managing the structures of computer systems as reconfiguration. The standard (classical) technology for reconfiguring computer systems with a reconstructible structure and programmable logic in case of failure of one of its resources includes the following main steps [5-7].

Step 1. Determination and analysis of the time and place of a resource failure; removal of the function (task) performed on this resource from the solution; transfer of the function (task) to another resource (with/without saving the obtained intermediate results).

Step 2. Exclusion of a failed resource from the configuration of computer systems with a reconstructible structure and programmable logic; an attempt to replace it with a backup (of the same type), or a backup of a different type with similar functionality.

Step 3. Exclusion of links to a failed resource, denial of access to it and for the failed resource itself – an attempt to restore it. In case if a high-priority function (task) has been solved on a failed resource, which, when transferred to other resources, begins to conflict with functions (tasks) assigned to this resource, then, depending on the service discipline, the execution of less priority functions (tasks) is interrupted; or simple removal from the solution. Thus, in the framework of the standard reconfiguration in case of failures and violations of the correct functioning of the corresponding node in the computer system in order to maintain the highest priority functions of the specified object or acceptable operating conditions, they “sacrifice” other functions or part of the operable elements. In some cases, this reconfiguration is called a “blind” reconfiguration, since during its implementation, as a rule, the following operations are not carried out: accounting and analysis of the current characteristics of tasks and functions performed in the computer system; analysis and assessment of the current state of the computer system as a whole; operative calculation, evaluation and analysis of the target and information-technical capabilities of the computer system for a reasonable redistribution of the functions of a computer system between its operable elements and subsystems. A prerequisite for the implementation of standard reconfiguration technology is the presence of a multi-level model of the computer system. operability. A possible graphical description of this model for one of the subsystems (structures) is shown in Figure 1. In Figure 1, the vertices of the graph are associated with the technical states (TSs) of a computer system. During operation, its elements can fall into various types of TSs: a functioning TS, a faulty TS, an operative TS, an inoperative TS, as well as correct and incorrect functioning. Figure 1 shows the main classes of the computer system TSs:

- the class of fully operative and functioning states:

$$S^{(u)} = \{S_{\alpha}^{(u)}\}, \quad \alpha = 1,\dots,N^{(u)}.$$

Elements of this class differ from each other in the level of accumulated deviations from the norm of performance parameters in certain elements of the computer system;

- the class of partially operative states with the *B*th level of efficiency and the *y*th number of accumulated deviations. At the same time, it is believed that with an increase in the index number, the *B*th level of the computer system operability will decrease:

$$S^{(p)} = \{S_{By}^{(p)}\}, \quad B = 1,\dots,N_1^{(p)},\ y = 1,\dots,N_2^{(p)}.$$

In the graph, operability levels are separated by horizontal dashed lines;

- the class of inoperative (failure) states of the computer system caused by the appearance of the *C*th type with the *N*th amount of losses:

$$S^{(o)} = \{S_{CN}^{(o)}\}, \quad C = 1,\dots,N_1^{(O)},\ N = 1,\dots,N_2^{(O)}.$$

In Figure, 1 S_L is the limit state, i.e. the state, in which all elements of the computer system are inoperative and the level of losses is maximum.

In this Figure, the arcs of the graph correspond to transitions of the computer system from one state to another. These transitions, as noted earlier, have a different nature: some are structurally functional, others are not structurally functional. There exists the following classification:

- D-transitions, which are caused by failures of elements of a computer system and its transition into a state with a lower level of performance or a higher amount of losses;
- A-transitions, which are caused by the achievement of the established threshold values of the accumulated deviations number;

- R-transitions, which are caused by partial or complete recovery of the computer system operability;
- C-transitions, which are caused by compensation of the accumulated deviations in the parameters of both the elements and the computer system as a whole.

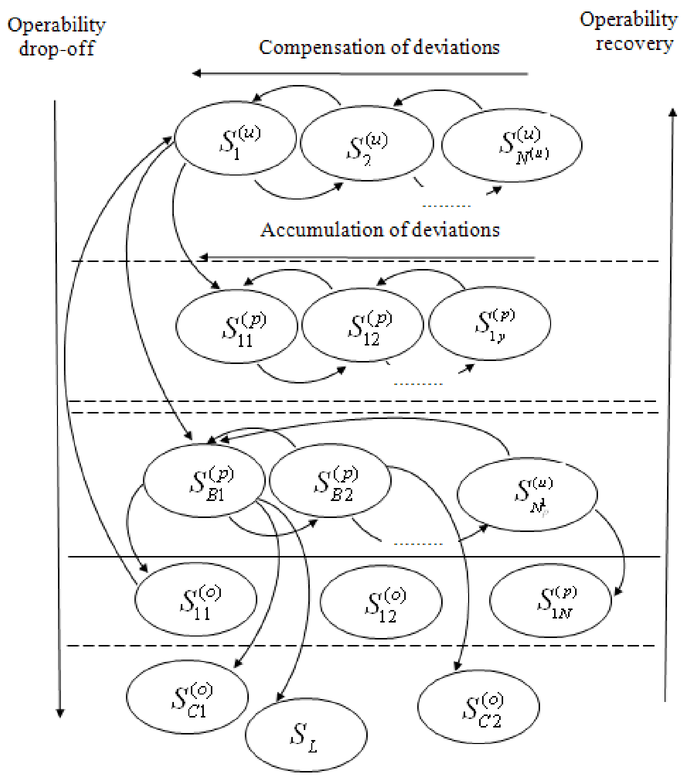

Fig. 1. The graph of evolution of computer system performance with a reconstructible structure and programmable logic

In Figure 1, D-transitions are depicted by arcs directed from top to bottom; A-transitions are given by horizontal arcs going from left to right; R-transitions are shown by arcs going from bottom to top; C-transitions are the arcs from right to left. The dashed line in Figure 1 marks the boundary between the operational and failure states of a computer system. Given the above, the dynamics of the computer system operability in the general case assumes the simultaneous implementation of the following processes of changing the technical state of both elements of the computer system and the system as a whole:

- degradation processes (D-processes);
- recovery processes (R-processes); - processes of accumulating parameter deviations (A-processes);

- compensation processes of accumulated parameter deviations (C-processes).

In this case, one of the goals of controlling the structural dynamics of computer systems with a reconstructible structure and programmable logic is to provide the highest possible level of operability of a computer system and its elements at every moment in time. This goal is achieved by two complementary processes. First, by influencing (controlling) the D-process (degradation) the possibility (probability) of transitions of the computer system to the least desirable states is eliminated or reduced. Second, it is done by using the organization (management) of the R-process (recovery) and the C-process (compensation). To implement these processes both in the computer system itself and in each of its elements (subsystems), a structural control loop is formed. Due to this loop, the scheduling processes are realized by reconfiguring computer systems with a reconstructible structure and programmable logic. Speaking about the possible classes of tasks for reconfiguring computer systems, one should, first of all, proceed from the so-called functional-structural approach (FSA) to the description of objects of any nature (including computer systems). Characteristic of FSA features are:

- considering the dialectical relationship of functions and structure of objects with the decisive role of the function in relation to the structure;
- a holistic approach to the analysis and synthesis of multi-level systems;
- considering material, energy and information links between elements of the system;
- considering the relationship of the investigated (created) system with the external environment.

The analysis of problems of reconfiguring computer systems taking into account the FSA allowed to identify the basic concepts and definitions that are shown in Figure 2. At the same time, considering various combinations of independent branches of the given morphological tree, we can obtain a kaleidoscopic set of reconfiguration tasks for computer systems with a reconstructible structure and programmable logic.

These systems operate in conditions of significant non-stationarity and uncertainty associated with changes in the content of goals and tasks, which computer systems with a reconstructible structure and programmable logic face, the influence of disturbing factors from the external environment and factors having a purposeful and/or unfocused nature. Moreover, in real life, disturbing environmental factors can be deterministic or unknown, described using probability-statistical or fuzzy-probable formalisms. In the process of reconfiguring computer systems with a reconstructible structure and programmable logic, the following main goals are pursued: preservation, implementation and restoration of the target and information-technological capabilities of the system. The implementation of reconfiguration can be carried out simultaneously in several directions.

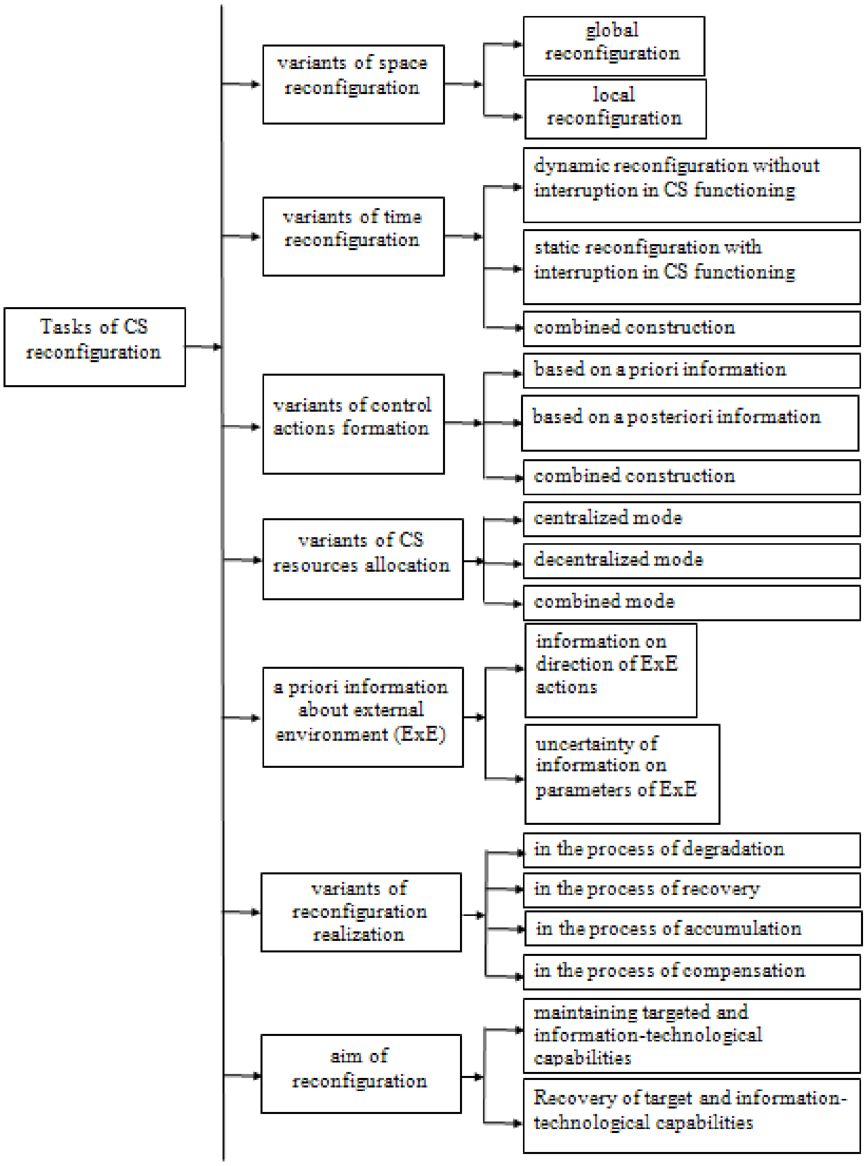

Fig. 2. The classification of reconfiguration tasks for computer systems with a reconstructible structure and programmable logic

The "vertical" direction implies the implementation of a reconfiguration of a computer system when it affects two complementary processes. The first one is the process of computer system degradation in order to eliminate or reduce the possibility of a computer system transferring to the least desirable states. The second one is the recovery process in order to partially or fully achieve an operable state. The "horizontal" direction is the impact on the process of accumulating deviations of FE parameters (or the computer system as a whole) and the impact on the process of compensating for accumulated deviations. In addition, the following can be considered as the most essential morphological features of the process of a computer system reconfiguration: space, time, resource allocation mode and the formation of control actions. According to the "space" feature, the variants for reconfiguring a computer system can be distinguished into local and global. A local variant of reconfiguration of a computer system deals with individual FEs, sets of FEs and/or individual subsystems of a computer system. In the case of global reconfiguration, the process relates generally to the entire computer system. The "time" classification feature is closely related to the state of the computer system and its objects during reconfiguration. Static reconfiguration is performed after the computer system has been interrupted. The main disadvantages of this type of reconfiguration are: - demand to stop the computer system, which entails not only the shutdown of the computer system, but also the shutdown of the associated service objects; - reconfiguration of the computer system should be carried out by external means since in these conditions the internal resources of the computer system cannot be used. The advantages of the dynamic reconfiguration, which is performed during the operation of a computer system, are the ability to maintain the operability of its service subsystems during the reconfiguration of a computer system and the ability to use its internal resources and reserves [8-10]. The development of control actions during reconfiguration of a computer system can be based on both a priori and a posteriori information about the behavior of individual FEs and subsystems of a computer system. As for the options for reallocating the resources of a computer system in the process of its reconfiguration, the following modes can be distinguished: centralized, decentralized, or a combination of these modes. In the centralized mode, the allocation of resources is carried out using a unified model that includes both the initial state of the computer system and all kinds of (most likely) intermediate states caused by destructive influences. In the decentralized mode, the selection of a resource allocation plan for the initial state of a computer system is carried out in such a way that subsequent reallocation caused by adverse effects will allow the rational use of computer system resources. The determination of optimal control programs for the main elements and subsystems of a computer system can be performed only after the list of functions and algorithms for information processing and control becomes known, which should be implemented in these elements and subsystems [11-16]. In turn, the distribution of functions and algorithms by elements and subsystems of a computer system depends on the structure and parameters of the laws of managing these elements and subsystems. The evolving situation is closely connected with the principle of the duality of the past and the future. This means that you can have knowledge about the past, but you cannot control it, and you can design the future without knowing it.

3 Algorithm for solving the problem of constructing a trajectory for structural reconfiguration of computer systems with a reconstructible structure and programmable logic

The task of constructing a trajectory for the structure reconfiguration during the degradation or recovery of a computer system with a reconstructible structure and programmable logic can be represented as the following optimization problem:

$$\sum_{j=0}^{N} F_{failure}(\vec{x}_{a_j}) \to \min_{\substack{\vec{x}_{a_j} \in X(\vec{x}_{a_{j-1}}) \\ \vec{x}_{a_0} = \vec{x}_0,\ \vec{x}_{aN} = \vec{x}_f, \\ \Psi_l(\vec{x}_{a_j}) \le 0,\ l=1,2,\ldots L \\ \{Q_{j1}, Q_{j2}, \ldots, Q_{jN}\} = \hat{Q}}} . \quad (1)$$

To solve the considered problem of structural reconfiguration of a computer system, we use a combined method, which includes the method of random guided search and cutting off unpromising trajectories for structural reconfiguration of a computer system using the "branch and bound" method. The random guided search in solving the problem of scheduling the structural reconfiguration of a computer system can be implemented as follows. When building a random sequence:

$$\mu_{\zeta}^{(k)} = [\vec{x}_{a_0}, \vec{x}_{a_1}^{(k)}, \vec{x}_{a_2}^{(k)}, \ldots, \vec{x}_{a_{N-1}}^{(k)}, \vec{x}_{a_N}] . \quad (2)$$

of the intermediate states of the trajectory for structural reconfiguration, we are in some intermediate structural state, which is characterized by the genome $\vec{x}_{a_j}^{(k)}$ from which we can go to one of the states of the next level of the computer system structure degradation through the failure of one of the operable FE sets: $\{Q_{i_1}, Q_{i_2}, \ldots, Q_{i_p}\} \subseteq Q$. Moreover, the values $a_r = F_{failure}(\vec{x}_{i_r}^{+}),\ r = 1, \ldots, p$ characterize the contributions to the structural failure upon failure of these FEs. In the general case, $a_r \in [-1, 1]$. To carry out the guided search, the interval [0,1] is divided into subintervals so that the coordinates of the ends of the subintervals form the sequence

$$0, \omega_1 = b_1, \omega_2 = \omega_1 + b_2, \ldots, \omega_p = \omega_{p-1} + b_p = 1 . \quad (3)$$

where $b_r = \dfrac{1 - a_r}{\Lambda}$; $\Lambda = \sum_{r=1}^{p} (1 - a_r)$. Here, the smaller the contribution a_r to the value of the structural failure index when the FE is removed $Q_{i_r} \in \hat{Q}$, the greater the length of the sub-interval b_r is assigned to this element, i.e. the element Q_{i_r} is "encouraged" by the large sub-interval. For random selection of the FE removed from the structure, a random number ζ is generated that is distributed uniformly in the interval from 0 to 1, which falls into a certain sub-interval $\xi \in (\omega_{r-1}, \omega_r)$ and thereby determines the selection of the FE removed from the structure of the computer system. Thus, in accordance with the constructed probabilistic hypothesis, a search for a global extremum is organized. In addition, in the vicinity of the global extremum, the density of randomly selected trajectories for a reconfigurable computer system will be the highest. To increase the efficiency of random guided search, it is proposed for each test to compare the intermediate values of the total structural failure indicator obtained on this trajectory of the computer system reconfiguration with the record value and stop the started test as soon as the intermediate value of this indicator exceeds the record one. If the specified number of tests has been carried out, then the calculation is considered complete. The calculation can be considered complete even if subsequent tests do not provide an improvement in the result. Using the proposed approach allows to guarantee the reception of the trajectory of the computer system structural reconfiguration in a sufficiently small vicinity of the optimal trajectory. At the same time, the process of finding a solution can be limited both by the time allocated for making a decision, by the productivity of the used computing tools, and by analyzing the nature of the process of taken decisions convergence to the optimum. The proposed approach can be implemented in the form of an algorithm consisting of the following steps.

Step 0. The initial state. We set the number of statistical tests M; the number of tests *d* that does not give an improvement in the result; current test k = 0; the record $F_{failure}^{record} = +\infty$.

Step 1. The beginning of the next test. List of deleted critical FEs $\hat{Q} = \{Q_{i_1}, Q_{i_2}, ..., Q_{i_N}\}$; intermediate total structural failure $F_{failure} = 0$.

Step 2. Random selection of the deleted FE. One of the FEs is randomly selected from the list $\hat{Q}$ in the manner described above, let it be Q_{i_r}. The intermediate structural state of the current trajectory is determined $\vec{x}_{a_j}^{(k)} (\vec{x}_{a_{j-1}}^{(k)} \to \vec{x}_{a_j}^{(k)})$.

Step #3. Analysis of the intermediate structural state $\vec{x}_{a_j}^{(k)}$. The structural failure $F_{failure}(\vec{x}_{a_j}^{(k)})$ and the change in the current total structural failure $F_{failure} = +F_{failure}(\vec{x}_{a_j}^{(k)})$ are calculated. If - $F_{failure} > F_{failure}^{record}$ ($F_{failure}^{record}$ is the current record), then proceed to **Step 5**.

Step 4. Analysis of the list of deleted FEs. FE Q_{i_r} is excluded from the list $\hat{Q}$. If $\hat{Q} = \varnothing$, then the test is over. Moreover, if $F_{failure} < F_{failure}^{record}$, then $F_{failure}^{record} = F_{failure}$. If $\hat{Q} = \varnothing$, go to **Step 2**.

Step 5. Analysis of test results. If the number of consecutive tests with no results $F_{failure} > F_{failure}^{record}$ is greater than *d*, then the procedure ends. If k> M, then the procedure ends. Otherwise, go to **Step 1**. The finiteness of the algorithm constructed according to this scheme follows from the finiteness of the set $Q = \{Q_{i_1}, Q_{i_2}, ..., Q_{i_N}\}$, the finiteness of the number of tests *M* and the sequential deletion of the FEs from the set $Q = \{Q_{i_1}, Q_{i_2}, ..., Q_{i_N}\}$.

4 Conclusions and recommendations

The analysis of the optimization problem features in constructing scenarios for the structural reconfiguration of computer systems with a reconstructible structure and programmable logic was carried out. The results have shown that methods and algorithms of the theory of search engine optimization of evolutionary type should be used to solve the considered problem. To solve the given problem, the method and the algorithm that implements this method are proposed, the novelty of which is the combined use of the random guided search method and the method of cutting off unpromising structural reconfiguration options for computer systems with a reconstructible structure and programmable logic such as "branch and bound". The proposed approach allows solving the optimization problems of constructing scenarios for the structural reconfiguration of computer systems with a reconstructible structure and programmable logic, both unconditional and conditional, that can be described using various structural and topological indicators. At the same time, modification of the proposed method allows the construction of a series of optimistic and pessimistic scenarios for the structural reconfiguration of computer systems with a reconstructible structure and programmable logic. The research has been carried out on the basis of the Educational and Scientific Laboratory of "Reconfigurable and Mobile Systems" at the Department of Electronic Computers in Kharkov National University of Radio Electronics.

References

1. Churyumov, G., Tkachov, V., Tokariev, V., Diachenko, V.: Method for Ensuring Survivability of Flying Ad-hoc Network Based on Structural and Functional Reconfiguration. In: XVIII International Scientific and Practical Conference «Information Technologies and Security» (ITS 2018), pp. 64-76. Kyiv, Ukraine (2018).
2. Tkachov, V., Tokariev, V., Dukh, Ya., Volotka, V.: Method of Data Collection in Wireless Sensor Networks Using Flying Ad Hoc Network. In 5th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology, pp 197-201, Kharkiv, Ukraine (2018).
3. Dodono,v A., Gorbachyk, O., Kuznietsova, M.: Management organization of mobile technical objects group / In: XVII International Scientific and Practical Conference «Information Technologies and Security» (ITS 2017), pp. 1-7. Kyiv, Ukraine (2017).
4. Dodonov, A., Gorbachyk, O., Kuznietsova, M.: Increasing the survivability of automated systems of organizational management as a way to security of critical infrastructures. In: XVIII International Scientific and Practical Conference «Information Technologies and Security» (ITS 2018), pp. 261-270. Kyiv, Ukraine (2018).
5. Barabash, O., Kravchenko, Y., Mukhin, V., Kornaga, Y., Leshchenko, O.: Optimization of Parameters at SDN Technologie Networks. International Journal of Intelligent Systems and Applications. 9 (9), 1-9 (2017).
6. Lemeshko, O., Yeremenko, O., Tariki, N.: Solution for the Default Gateway Protection within Fault-Tolerant Routing in an IP Network. International Journal of Electrical and Computer Engineering Systems. 8 (1), 19–26 (2017).

7. Bielievtsov, S., Ruban, I., Smelyakov, K., Sumtsov, D.: Network technology for transmission of visual information // In: XVIII International Scientific and Practical Conference «Information Technologies and Security» (ITS 2018), pp. 161-275. Kyiv, Ukraine (2018).
8. Pavlov, A.N.: Metodologicheskiye osnovy resheniya problemy planirovaniya struktur-no-funktsional'noy rekonfiguratsii slozhnykh ob"yektov. Izvestiya Vysshikh Uchebnykh Zavedeniy. Priborostroyeniye, 55 (11), 7-13.
9. Dodonov, A.G., Kuznetsova, M.G., Gorbachik, Ye.S.: Vvedeniye v Teoriyu Zhivuchesti Vychis-litel'nykh Sistem, Nauk. Dumka, Kyiv (1990).
10. Dodonov, A.G., Lande, D.V.: Zhivuchest' Informatsionnykh Sistem. Nauk. Dumka, Kyiv (2011).
11. Dodonov, A.G., Kuznetsova, M.G.: O nekotoroj strategii rekonfiguracii zhivuchih vychislitelnyh sistem. Gibridnye vychislitelnye mashiny i kompleksy, 11, 31-35 (1988).
12. Dodonov, A.G., Kuznetsova, M.G.: Raspredelenie resursov v zhivuchih vychislitelnyh sistemah. Gibridnye vychislitelnye mashiny i kompleksy, 7, 20-23 (1984).
13. Westmark, V. R.: A definition for information system survivability. In 37th Annual Hawaii International Conference on System Sciences, p. 10 (2004).
14. Romashkova, O.N., Dedova Ye.V.: Zhivuchest' besprovodnykh setey svyazi v usloviyakh chrezvychaynoy si-tuatsii. T-Comm: Telekommunikatsii i Transport, 6, 40-43 (2014).
15. Lande, D.V.: Metodi pidvishennya zhivuchosti informacijnoyi skladovoyi korporativnih informacijno-analitichnih sistem pidtrimki prijnyattya rishen. Reyestraciya, zberigannya i obrobka danih, 14 (2), 48-58 (2012).
16. Dodonov, A.G., Lande, D.V., Putyatin V.G. Informacijni potoki v globalnih komp'yuternih merezhah. Naukova dumka, Kyiv (2009).