DOI: https://doi.org/10.15276/aait.09.2026.25
UDC 004.05

A generalized method for constructing failure-behavior models of fault-tolerant multiprocessor systems with inter-subsystem redundancy

Vitaliy A. Romankevich[1)]
ORCID: https://orcid.org/0000-0003-4696-5935; zavkaf@scs.kpi.ua. Scopus Author ID: 57193263058
Kostiantyn V. Morozov[1)]
ORCID: https://orcid.org/0000-0003-0978-6292; mcng@ukr.net. Scopus Author ID: 57222509251
Alexei M. Romankevich[1)]
ORCID: https://orcid.org/0000-0001-5634-8469; romankev@scs.kpi.ua. Scopus Author ID: 6602114176
Daniil V. Halytsky[1)]
ORCID: https://orcid.org/0009-0004-4421-3443; zipper135401@gmail.com. Scopus Author ID: 58553487600
Lefteris Zacharioudakis[2)]
ORCID: https://orcid.org/0000-0002-9658-3073; l.zacharioudakis@nup.ac.cy. Scopus Author ID: 57422876200
[1)] National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, 37, Peremogy Ave. Kyiv, 03056, Ukraine
[2)] Neapolis University Pafos, 2, Danais Ave. Pafos, 8042, Cyprus

ABSTRACT

Relevance. The evaluation of reliability characteristics of fault-tolerant multiprocessor systems is an important task, particularly during their analysis and design. GL-models of system failure behavior can be used for this purpose. For systems with inter-subsystem redundancy, such models should take into account the possibility of using the processors of a donor subsystem both when the subsystem remains operational and after it loses operability. Existing approaches do not provide a complete representation of these operating modes. **Purpose.** To develop a generalized method for constructing GL-models of the failure behavior of fault-tolerant multiprocessor systems with inter-subsystem redundancy. **Objective.** To formalize the principles of constructing GL-models of a donor subsystem taking into account different modes of processor utilization, to develop an algorithm implementing the proposed method, and to perform experimental verification of the correctness of the constructed models. **Methods.** The study employs the GL-model formalism, methods for constructing basic and auxiliary GL-models, the formation of extended state vectors, and the analysis of graph connectivity. **Scientific novelty.** A generalized method for constructing GL-models of fault-tolerant multiprocessor systems with inter-subsystem utilization of reserve processors is proposed. Unlike existing approaches, the proposed method takes into account the possibility of using the processors of a donor subsystem both when it remains operational and after it loses operability, when its operational processors may still be used to support other subsystems. **Practical significance.** The proposed method can be used for constructing GL-models of complex fault-tolerant multiprocessor systems and for the subsequent evaluation of their reliability characteristics. **Results.** A method and an algorithm for constructing GL-models of systems with inter-subsystem redundancy have been developed. The proposed approach is based on constructing auxiliary models that describe the sets of reserve and operational processors of a donor subsystem and on forming extended state vectors for constructing GL-models of recipient subsystems. Representative examples are considered for systems with limitations on the number of reserve resources and for systems employing sliding redundancy. Experimental studies confirmed the correctness of the constructed models. It is shown that the complexity of the edge-function expressions of the resulting GL-models is comparable to that of the corresponding basic models with equivalent fault-tolerance parameters. **Conclusions.** The proposed method provides a formal representation of inter-subsystem redundancy within GL-models and can be integrated with other approaches to constructing failure-behavior models of fault-tolerant multiprocessor systems. The experimental results confirm the correctness of the proposed method and demonstrate the feasibility of its application in the analysis and design of complex fault-tolerant multiprocessor systems.

Keywords: Fault-tolerant multiprocessor systems; GL-models; inter-subsystem redundancy; reliability analysis; failure behavior; reserve resources

For citation: **Romankevich V. A., Morozov K. V., Romankevich A. M., Halytsky D. V., Zacharioudakis L. “A generalized method for constructing failure-behavior models of fault-tolerant multiprocessor systems with inter-subsystem redundancy”. *Applied Aspects of Information Technology.* 2026; Vol. 9 No. 3: 366–383. DOI: https://doi.org/10.15276/aait.09.2026.25**

INTRODUCTION

Modern technical systems (TS) are typically characterized by a high complexity of the processes implemented within them and require correct and reliable control. Assigning this function entirely or predominantly to a human operator is undesirable for several reasons, including the need to minimize the influence of the human factor and to reduce the workload associated with performing monotonous

© Romankevich V., Morozov K., Romankevich A., Halytsky D., Zacharioudakis L., 2026

This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/deed.uk)

and routine operations. In this regard, in modern TS the control functions are usually assigned to a specialized system – the control system (CS). The CS collects information from various sensors, processes the obtained data according to a specified algorithm, and generates control signals for the actuating units of the TS [1], [2], [3].

A failure of the control system may lead to incorrect operation of the technical system as a whole, which is unacceptable in view of the requirements for its safe and reliable operation. This is particularly relevant for critical application systems (CAS), whose failure may cause significant material losses and pose a threat to human life and health [4], [5], [6]. In this regard, increased requirements for reliability and fault tolerance are imposed on the control systems of such technical systems [7], [8], [9], [10].

In addition, it should be noted that the control of complex technical systems, most of which belong to critical application systems, often requires significant computational resources. The implementation of control systems for such TS based on fault-tolerant multiprocessor systems (FTMS), which contain a large number of processors (in some cases dozens or even hundreds) and, due to the use of various types of redundancy, are capable of continuing failure-free operation in the event of failures of individual components, makes it possible to simultaneously solve two key tasks. On the one hand, this ensures the achievement of the required level of computational performance, and on the other hand, it guarantees the necessary level of reliability through the implementation of fault-tolerance mechanisms.

During the design of fault-tolerant multiprocessor control systems, the problem of evaluating the reliability parameters of the corresponding system inevitably arises. This is necessary, in particular, to confirm its compliance with the specified requirements and to identify bottlenecks and potential directions for its improvement. Given the architectural complexity and the large scale of some fault-tolerant multiprocessor systems used in control applications, the task of such evaluation is considerably challenging.

LITERATURE REVIEW

The evaluation of reliability parameters of fault-tolerant multiprocessor systems can be carried out using various approaches, which can be divided into two main groups [11]. The first group includes methods based on the construction of analytical expressions that allow the required reliability indicators of the system to be calculated directly [12], [13], [14], [15]. The advantage of such methods is the possibility of obtaining exact values of reliability parameters for systems with a specified structure and known characteristics of their elements. At the same time, their disadvantage lies in their limited universality: for systems with a new or complex structure, there is often a need to develop specialized computational procedures. In addition, combining different analytical approaches with each other is usually complicated.

The second group includes methods based on conducting statistical experiments with models of the failure behavior of fault-tolerant multiprocessor systems. Such approaches are characterized by relative universality, since they can be applied to systems of arbitrary structure provided that an appropriate model of their behavior is available [16], [17], [18]. At the same time, the evaluation of reliability parameters in this case has a statistical nature and is performed with limited accuracy, which is determined, in particular, by the number of experiments conducted.

As models of the failure behavior of fault-tolerant multiprocessor systems, it is advisable to use GL-models [18]. Such a model is represented as an undirected graph in which each edge corresponds to a certain Boolean edge function. The arguments of these functions are the elements of the system state vector – a Boolean vector in which each component corresponds to the state of an individual processor: 1 indicates that the processor is operational, and 0 indicates that the processor has failed. Thus, each state vector corresponds to a specific combination of states of all processors in the system. If the edge function for a given state vector takes the value 0, the corresponding edge is considered absent in the graph. The connectivity of the model graph for a given state vector corresponds to the operable state of the system, whereas the loss of connectivity indicates its failure.

For systems composed of several subsystems, the overall state vector is formed by concatenating the subvectors corresponding to each subsystem, where each subvector represents the states of processors within the corresponding subsystem.

For the evaluation of reliability parameters, sets of system state vectors are generated. In the general case, these vectors may be obtained randomly, and the accuracy of the estimates depends on the number of experiments (i.e., the number of generated vectors). For maximum accuracy, exhaustive enumeration of all possible state vectors can be used. At the same time, more efficient estimation can be achieved by employing specialized pseudo-random generators that provide improved coverage of the state space for a fixed number of experiments [18].

Among fault-tolerant multiprocessor systems, it is reasonable to distinguish a separate class – basic systems, whose state in the failure behavior is determined exclusively by the number of operational processors. A basic fault-tolerant multiprocessor system containing n processors and remaining operable provided that no more than m of them have failed, as well as the corresponding GL-model, will be denoted as $K(m, n)$. Such systems are also known as k-out-of-n systems [19], [20]. In particular, it can be shown that the system $K(m, n)$ is equivalent to an $(m + 1)$-out-of-n:G system (with respect to the operability condition) and to an $(n – m)$-out-of-n:F system (with respect to the failure condition).

Systems whose failure behavior cannot be reduced to a dependence solely on the number of operational processors belong to the class of non-basic systems. For such systems, situations are possible in which, for the same number of failures, some combinations of failed processors lead to the preservation of operability, whereas others result in system failure. This class includes, in particular, consecutive-k-out-of-n [12], [13], [21], [22], [23], [24] consecutive-k_c-out-of-n [25], [26], consecutive-k_r-out-of-n_r [27], consecutive-(k, l)-out-of-n [28], (r, s)-out-of-(m, n) [29], [30], [31], consecutive-k-out-of-r-from-n [32], [33], consecutive-k-within-m-out-of-n [34], [35], m-consecutive-k-out-of-n [25], [26], [36], [37] m-consecutive-k,l-out-of-n [38], [39], (n, f, k) [40], [41], [42] $\langle n, f, k\rangle$ [41], [42], [43] as well as systems with more complex structures of dependencies among elements [44].

Known methods for constructing GL-models are primarily oriented toward basic fault-tolerant multiprocessor systems. For non-basic systems, the construction of models is usually performed by modifying the corresponding auxiliary basic models [44].

A GL-model of a basic fault-tolerant multiprocessor system can be constructed, in particular, using the method described in [44]. The model $K(m, n)$, where $n \geq 1$ and $1 \leq m \leq n$, constructed according to this method, is based on a cycle graph with $\varphi(n, m)$ edges,
where

$$\varphi(n,m) = n - m + 1. \qquad (1)$$

In addition, one of the properties of such models is that for any state vector containing l zero components, the model loses $\psi(m, l)$ edges, where

$$\psi(m,l) = \begin{cases} 0, & l < m, \\ l - m + 1, & l \geq m. \end{cases} \qquad (2)$$

Such GL-models are called MLE-models (Minimum Lost Edges), since for state vectors with $m + 1$ zero components they lose the minimum number of edges whose removal leads to a loss of connectivity of the graph (namely, two edges).

The construction of the expressions of the edge functions of an MLE-model is performed recursively based on the expressions of the edge functions of auxiliary MLE-models of smaller dimension. The system state vector is divided into two non-empty subvectors of sizes n_1 and n_2, where $n_1 + n_2 = n$, which are used as input vectors for the corresponding auxiliary models. In the general case, the values of n_1 and n_2 may be arbitrary; however, in practice the division is usually performed into two equal or nearly equal parts.

The set of edge functions of the model $K(m, n)$ is formed as the union of three subsets:

1) the set of edge functions of the auxiliary MLE-model $K_1(m, n_1)$ provided that $n_1 \geq m$;

2) the set of functions of the form

$$f = \bigwedge_{f_1 \in F_1} f_1 \vee \bigwedge_{f_1 \in F_2} f_2,$$

where F_1 is the set of edge functions of the auxiliary MLE-model $K_1(m – i, n_1)$, F_2 is the set of edge functions of the auxiliary MLE-model $K_2(i, n_2)$, $i = a, a + 1, \ldots, b$, $a = \max(1, m – n_1)$, $b = \min(m – 1, n_2)$, provided that $m \geq 2$;

3) the set of edge functions of the auxiliary MLE-model $K_2(m, n_2)$ provided that $n_2 \geq m$.

The recursive construction process continues until the boundary level of recursion is reached – models of type $K(1, 1)$, which contain a single edge function of the form $f = x_i$, where x_i is the corresponding component of the system state vector.

The described approach is effective for constructing GL-models of basic systems. At the same time, the problem of constructing GL-models for non-basic fault-tolerant multiprocessor systems, in particular those whose failure behavior is determined by more complex combinations of processor states, requires separate consideration.

PROBLEM STATEMENT

An important class of fault-tolerant multiprocessor systems consists of systems composed of several subsystems [44]. Such systems generally belong to the class of non-basic systems, although some of their subsystems may operate as basic ones. In a number of cases, such fault-tolerant multiprocessor systems are resilient not only to failures of individual processors but also to failures of entire subsystems. In addition, in order to improve reliability characteristics, the system may be organized so that, in the presence of redundant resources, one subsystem supports the operability of another, in particular by partially taking over its functions and acting as a *donor* for the *recipient* subsystem.

In [44], a method for constructing GL-models for systems in which subsystems interact in a donor-recipient mode is proposed. Within this approach, it is assumed that the *donor* subsystem always remains operable, which limits the applicability of the method.

However, in some fault-tolerant multiprocessor systems the possibility of failure of the *donor* subsystem is allowed. In this case, even after the loss of its operability, the subsystem may still contain operational processors that can be used to maintain or restore the operability of the *recipient* subsystem. Existing methods for constructing GL-models do not take into account this feature of the functioning of the *donor* subsystem.

Thus, the problem arises of developing a GL-model capable of correctly representing the behavior of the *donor* subsystem in the general case, namely: the use of its redundant resources when operability is preserved and the use of all its operational processors when it fails.

RESEARCH AIM AND OBJECTIVES

The aim of this work is to develop a method for constructing GL-models of fault-tolerant multiprocessor systems with inter-subsystem redundancy, including *donor* subsystems, taking into account the possibility of using their processors both when operability is preserved (with the involvement of redundant resources) and when the subsystem fails (with the use of all processors that remain operational).

The developed model can be used as a component of more general GL-models, in particular in a manner similar to the *donor* subsystem model described in [44], or for forming the state vector of sliding redundancy processors.

To achieve the stated aim, the following objectives are formulated in this work:

1) to formalize the principles for constructing a GL-model of a *donor* subsystem, taking into account the possibility of using its processors both when operability is preserved and when a failure occurs;

2) to develop an algorithm for implementing the proposed method;

3) to carry out experimental verification of the correctness of the GL-models constructed according to the developed method.

METHOD FOR CONSTRUCTING A GL-MODEL FOR DONOR-SUBSYSTEM

A system composed of several subsystems is considered. One of them will hereafter be referred to as the *donor* subsystem. When necessary, it can provide the resources of its processors that are not used for executing its own tasks in order to restore or maintain the operability of other *recipient* subsystems.

Let the *donor* subsystem contain n_d processors and be tolerant to the failure of no more than m_d of them. Its failure behavior is described by the GL-model $K(m_d, n_d)$. Such a model may be implemented as an MLE-model or as a basic GL-model constructed by other methods.

Let $\kappa_d \in \{0,1\}$ denote a Boolean variable determined by the specified GL-model that represents the state of the *donor* subsystem: $\kappa_d = 1$ corresponds to its operable state, whereas $\kappa_d = 0$ corresponds to the failure state.

Let l_d denote the number of zero components in the state vector of the *donor* subsystem. If $l_d \le m_d$ (i.e., the *donor* subsystem is in an operable state), the resources of its *redundant* processors may be used to maintain or restore the operability of the recipient subsystems. In [44], it is shown that the number of such processors can be determined using the dual model $K^*(n_d - m_d + 1, n_d)$, obtained from the MLE-model $K(n_d - m_d + 1, n_d)$ by replacing its edge functions with their dual ones. This model will be denoted as $R_d(m_d, n_d)$.

Let $\mathbf{r}_d$ denote the Boolean vector of the values of the edge functions of the model $R_d(m_d, n_d)$ for a given state vector. Then the number of ones in the vector $\mathbf{r}_d$ is equal to the number of *redundant* processors of the *donor* subsystem.

If $l_d \ge m_d + 1$, the subsystem enters a failure state; however, its operational (orphan) processors may also be used to support other recipient subsystems. To determine them, let us introduce the MLE-model $K(m_d + 1, n_d)$, which will be denoted as $O_d(m_d, n_d)$. According to (1), this model contains $n_d - m_d$ edges. According to (2), for state vectors with $l_\mathrm{d} \geq m_\mathrm{d} + 1$ zero components, the model loses $l_\mathrm{d} - m_\mathrm{d}$ edges; that is, exactly this number of its edge functions takes the value zero. Thus, the number of edge functions that have the value 1 is equal to

$$(n_\mathrm{d} - m_\mathrm{d}) - (l_\mathrm{d} - m_\mathrm{d}) = n_\mathrm{d} - l_\mathrm{d},$$

which is equal to the number of operational *orphan* processors of the *donor* subsystem.

Let $\mathbf{o}_\mathrm{d}$ denote the corresponding Boolean vector of the values of the edge functions of the model $O_\mathrm{d}(m_\mathrm{d}, n_\mathrm{d})$. Then the number of ones in $\mathbf{o}_\mathrm{d}$ is equal to the number of *orphan* processors.

Let us construct a Boolean vector whose unit components correspond to the processors of the *donor* subsystem available for use in maintaining or restoring the operability of the *recipient* subsystems. Obviously, when the *donor* subsystem remains operable, the number of such processors is equal to the number of *redundant* processors, whereas in the case of its failure it is equal to the number of *orphan* processors.

Let $\mathbf{r}_\mathrm{d} = (g_1, \dots, g_{k_\mathrm{r}})$ denote the vector of the values of the edge functions of the model $R_\mathrm{d}(m_\mathrm{d}, n_\mathrm{d})$, and $\mathbf{o}_\mathrm{d} = (h_1, \dots, h_{k_\mathrm{o}})$ the vector of the values of the edge functions of the model $O_\mathrm{d}(m_\mathrm{d}, n_\mathrm{d})$, where k_r and k_o are the numbers of edge functions in the corresponding models. Let $k = \max(k_\mathrm{r}, k_\mathrm{o})$.

Construct the vectors $\mathbf{r} = (r_1, r_2, \dots, r_k)$ and $\mathbf{o} = (o_1, o_2, \dots, o_k)$ by extending the vectors $\mathbf{r}_\mathrm{d}$ and $\mathbf{o}_\mathrm{d}$ to length k with zero components, that is:

$$r_i = \begin{cases} g_i, & 1 \leq i \leq k_\mathrm{r}, \\ 0, & k_\mathrm{r} < i \leq k, \end{cases}$$

$$o_i = \begin{cases} h_i, & 1 \leq i \leq k_\mathrm{o}, \\ 0, & k_\mathrm{o} < i \leq k. \end{cases}$$

Since the extension is performed using zero components, the number of ones in the vectors $\mathbf{r}$ and $\mathbf{r}_\mathrm{d}$, as well as in $\mathbf{o}$ and $\mathbf{o}_\mathrm{d}$, coincides.

Let us define the vector $\mathbf{v} = (v_1, \dots, v_k)$, whose components are given by the relation

$$v_i = \kappa_\mathrm{d} r_i \vee \bar{\kappa}_\mathrm{d} o_i, i = 1, \dots, k.$$

Then

$$\mathbf{v} = \begin{cases} \mathbf{r}, & \kappa_\mathrm{d} = 1, \\ \mathbf{o}, & \kappa_\mathrm{d} = 0. \end{cases}$$

That is, if $\kappa_\mathrm{d} = 1$, the number of ones in $\mathbf{v}$ is equal to the number of *redundant* processors, whereas if $\kappa_\mathrm{d} = 0$, it is equal to the number of *orphan* processors. Thus, the number of ones in $\mathbf{v}$ is equal to the number of processors of the *donor* subsystem available for maintaining or restoring the operability of the *recipient* subsystems.

Note. If $\kappa_\mathrm{d} = 0$ (i.e., $l_\mathrm{d} \geq m_\mathrm{d} + 1$ and the *donor* subsystem is in a failure state), then for the model $R_\mathrm{d}(m_\mathrm{d}, n_\mathrm{d})$ all edge functions take the value zero: $g_i = 0$ for all $i = 1, 2, \dots, k_\mathrm{r}$. Accordingly, $r_i = 0$ for all $i = 1, 2, ,\dots, k$, and the expression for the components of the vector $\mathbf{v}$ can be simplified to

$$v_i = r_i \vee \bar{\kappa}_\mathrm{d} o_i, i = 1, \dots, k.$$

Proof. The model $R_\mathrm{d}(m_\mathrm{d}, n_\mathrm{d})$ is defined as the dual of the MLE-model $K(m^*_\mathrm{d}, n_d)$, where $m^*_\mathrm{d} = n_\mathrm{d} - m_\mathrm{d} + 1$. Let $\mathbf{x}_\mathrm{d}$ be the state vector of the *donor* subsystem with l_d zero components, and $\mathbf{y}_\mathrm{d} = \mathbf{x}^*_\mathrm{d}$ be its bitwise complement. Then the number of zero components l^*_d in $\mathbf{y}_\mathrm{d}$ is equal to the number of ones in $\mathbf{x}_\mathrm{d}$, that is, $l^*_\mathrm{d} = n_\mathrm{d} - l_\mathrm{d}$. If $l_\mathrm{d} \geq m_\mathrm{d} + 1$, then $l^*_\mathrm{d} = n_\mathrm{d} - l_\mathrm{d} \leq n_\mathrm{d} - (m_\mathrm{d} + 1) = = n_\mathrm{d} - m_\mathrm{d} - 1$. Moreover, it is obvious that $n_\mathrm{d} - m_\mathrm{d} - 1 < n_\mathrm{d} - m_\mathrm{d} + 1 = m^*_\mathrm{d}$. Thus, for the MLE-model $K(m^*_\mathrm{d}, n_d)$ on the vector $\mathbf{y}_\mathrm{d}$ the condition $l^*_\mathrm{d} < m^*_\mathrm{d}$ holds, and according to (2) it does not lose any edge; none of the edge functions takes the value zero, that is, $f_i(\mathbf{y}_\mathrm{d}) = 1$ for all i.

By the definition of duality of edge functions, we have $f_i^*(\mathbf{x}_\mathrm{d}) = \bar{f}_i(\mathbf{x}^*_\mathrm{d}) = \bar{f}_i(\mathbf{y}_\mathrm{d})$. Therefore, from $f_i(\mathbf{y}_\mathrm{d}) = 1$it follows that $f_i^*(\mathbf{x}_\mathrm{d}) = 0$ for all i. ■

It should be noted that the number of processors of the *donor* subsystem that can be used to maintain or restore the operability of the *recipient* subsystems may be additionally limited by a specified value h. In [44], it is proved that to implement such a limitation it is sufficient to construct an auxiliary GL-model $L(h, t)$, where t is the number of components of the vector $\mathbf{v}$. As the model L, an MLE-model of the form $K(t - h + 1, t)$ is used. The vector $\mathbf{v}$, formed at the previous stage, is supplied as the input to the model $L(h, t)$. Let $\mathbf{v}'$ denote the vector of the values of its edge functions. Then the number of ones in $\mathbf{v}'$ is equal to the number of processors available for use taking into account the limitation h.

Thus, the vector $\mathbf{v}$ (or $\mathbf{v}'$ when the limitation h is introduced) formalizes the set of processors of the *donor* subsystem available for use by other subsystems. Subsequently, this vector is used to construct GL-models of the *recipient* subsystems.

In particular, as shown in [44], in the case of a single *recipient* subsystem containing n_r processors and tolerant to the failure of no more than m_r of them, the corresponding GL-model can be specified as a basic model (in particular, an MLE-model) of the form $K_r(m_r + t, n_r + t)$, where t is the number of components of the vector $\mathbf{v}$. The input vector for this model is an extended state vector formed by concatenating the state vector of the *recipient* subsystem (i.e., the corresponding subset of components of the overall system state vector) and the vector $\mathbf{v}$.

If there are several *recipient* subsystems and the resources of the *donor* subsystem are used according to the principle of sliding redundancy, the other one method can be applied. According to this approach, a set of N *recipient* subsystems is considered, each of which contains n_j processors and is tolerant to the failure of no more than m_j of them, where $j = 1, 2, \dots, N$. For each subsystem, an auxiliary MLE-model of the form $K_j(m_j + 1, n_j)$ is constructed. Based on the values of the edge functions of these models, Boolean vectors $\mathbf{v}_j$ are formed that represent the number of failed processors requiring compensation.

Next, by concatenating the vectors $\mathbf{v}_j$, $j = 1, 2, \dots, N$, and the vector $\mathbf{v}$ (which corresponds to the state of the sliding redundancy processors), a combined Boolean vector of length s is formed. This vector is used as the input for a basic GL-model of the form $K(t, s)$, where t is the number of sliding redundancy processors, i.e., the length of the vector $\mathbf{v}$. The model constructed in this way represents the condition of simultaneous operability of all *recipient* subsystems.

Thus, the proposed approach is not limited to the construction of a GL-model of the *donor* subsystem. In addition to this model, the method involves the construction of auxiliary GL-models that describe the sets of *redundant* and operational (*orphan*) processors, as well as the formation of extended state vectors based on their outputs. These vectors serve as input data for constructing GL-models of *recipient* subsystems or systems with sliding redundancy. Therefore, the proposed method defines a unified framework for constructing a class of interrelated GL-models that together represent the failure behavior of the overall system.

ALGORITHM FOR CONSTRUCTING A GL-MODEL FOR DONOR-SUBSYSTEM

Based on the proposed method, the following algorithm for constructing a GL-model of the *donor* subsystem can be formulated.

1. **Construction of the donor subsystem state model.** Construct the basic GL-model $K_d(m_d, n_d)$ that describes the failure behavior of the *donor* subsystem. Denote the Boolean result (the subsystem state) determined by this model as $\kappa_d \in \{0,1\}$.

2. **Construction of the model for redundant processors.** Construct the auxiliary GL-model $R_d(m_d, n_d)$ corresponding to the case where the *donor* subsystem remains operable and used to determine the *redundant* processors. As $R_d(m_d, n_d)$, use the dual model of the form $K^*(n_d - m_d + 1, n_d)$. Compute the vector of the values of its edge functions $\mathbf{r}_d = (g_1, \dots, g_{k_r})$.

3. **Construction of the model for orphan processors.** Construct the auxiliary GL-model $O_d(m_d, n_d)$ corresponding to the case of failure of the *donor* subsystem and used to determine the *orphan* processors. As $O_d(m_d, n_d)$, use the MLE-model of the form $K(m_d + 1, n_d)$. Compute the vector of the values of its edge functions $\mathbf{o}_d = (h_1, \dots, h_{k_o})$.

4. **Formation of the vector of available processors.** Let $k = \max(k_r, k_o)$. Extend the vectors $\mathbf{r}_d$ and $\mathbf{o}_d$ with zero components to length k, obtaining $\mathbf{r} = (r_1, \dots, r_k)$ and $\mathbf{o} = (o_1, \dots, o_k)$. Form the vector $\mathbf{v} = (v_1, \dots, v_k)$ using the formula $v_i = \kappa_d r_i \vee \bar{\kappa}_d o_i$, $i = 1, 2, \dots, k$, where r_i and o_i are the corresponding values of the edge functions of the models R_d and O_d (extended with zeros where necessary). **Note.** Since $r_i = 0$ when $\kappa_d = 0$ for all $i = 1, 2, \dots, k_r$, as proved in the previous section, the formula can be simplified to $v_i = r_i \vee \bar{\kappa}_d o_i$.

5. **(Optional) Limiting the number of available processors.** If it is necessary to limit the number of available processors to the value h, construct the auxiliary GL-model $L(h, k)$ as an MLE-model of the form $K(k - h + 1, k)$. Provide the vector $\mathbf{v}$ as the input to the model $L(h, k)$ and form the vector $\mathbf{v}'$ from the values of its edge functions. Use $\mathbf{v}'$ instead of $\mathbf{v}$.

6. **Further use of the result.** Use the vector $\mathbf{v}$ (or $\mathbf{v}'$) to construct the GL-model of the *recipient* subsystem according to [44] or to form the state vector of the sliding redundancy processors.

EXAMPLES AND EXPERIMENTAL RESULTS

Example 1. As an example, consider a system consisting of two subsystems, hereafter referred to as *Subsystem 1* and *Subsystem 2*. *Subsystem 1* contains $n_1 = 6$ processors and is tolerant to the failure of no more than $m_1 = 2$ of them. *Subsystem 2* contains $n_2 = 8$ processors and is tolerant to the failure of no more than $m_2 = 3$ processors (Fig. 1). It is assumed that the resources of *Subsystem 2* can be used to maintain or restore the operability of *Subsystem 1*: when *Subsystem 2* remains operable – through its *redundant* processors, and in the case of its failure – through its operational *orphan* processors. Thus, *Subsystem 2* acts as the *donor*, whereas *Subsystem 1* is the *recipient* subsystem. Next, the corresponding GL-models of the failure behavior of both subsystems will be constructed.

The elements of the state vector corresponding to the processors of *Subsystem 1* will be denoted as $x_1^1, x_2^1, \dots, x_6^1$, and the elements corresponding to the processors of *Subsystem 2* as $x_1^2, x_2^2, \dots, x_8^2$.

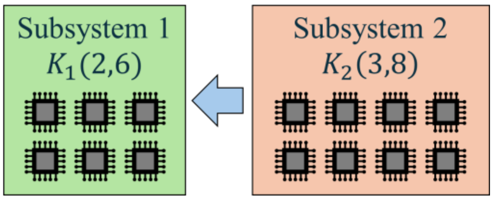

Fig. 1. **FTMS from *Example 1***
***Source:* compiled by the authors**

First, let us construct the MLE-model $K_2(3, 8)$, which describes the failure behavior of *Subsystem 2*. Since $n_2 = 8$ and $m_2 = 3$, this model is constructed on a cycle graph with $n_2 – m_2 + 1 = 6$ edges (Fig. 2). Its edge functions have the form:

$$f_1^2 = x_1^2 \vee x_2^2 \vee x_3^2 x_4^2;$$
$$f_2^2 = x_1^2 x_2^2 \vee x_3^2 \vee x_4^2;$$
$$f_3^2 = (x_1^2 \vee x_2^2)(x_1^2 x_2^2 \vee x_3^2 x_4^2)(x_3^2 \vee x_4^2) \vee$$
$$\vee\, x_5^2 x_6^2 x_7^2 x_8^2;$$
$$f_4^2 = x_1^2 x_2^2 x_3^2 x_4^2 \vee$$
$$\vee \left(x_5^2 \vee x_6^2\right)\left(x_5^2 x_6^2 \vee x_7^2 x_8^2\right)(x_7^2 \vee x_8^2);$$
$$f_5^2 = x_5^2 \vee x_6^2 \vee x_7^2 x_8^2;$$
$$f_6^2 = x_5^2 x_6^2 \vee x_7^2 \vee x_8^2.$$

The Boolean result obtained using the GL-model $K_2(3, 8)$, i.e., the state of *Subsystem 2*, will be denoted as $\kappa_2 \in \{0,1\}$.

Next, let us construct a GL-model for representing the redundant processors of *Subsystem 2* in the case where it remains operable. According to the described method, this model $R_2(3, 8)$ is defined as the dual of the MLE-model $K(n_2 – m_2 + 1, n_2) = K(6, 8)$. That is, $R_2(3, 8) = K^*(6, 8)$. It is based on a cycle graph with $8 – 6 + 1 = 3$ edges (Fig. 2), and its edge functions have the form:

$$g_1 = x_1^2 x_2^2 x_3^2 x_4^2 \wedge$$
$$\wedge \left(x_5^2 x_6^2 \vee (x_5^2 \vee x_6^2)(x_7^2 \vee x_8^2) \vee x_7^2 x_8^2\right);$$
$$g_2 = (x_1^2 x_2^2 (x_3^2 \vee x_4^2) \vee (x_1^2 \vee x_2^2) x_3^2 x_4^2) \wedge$$
$$\wedge \left(x_5^2 x_6^2 (x_7^2 \vee x_8^2) \vee \left(x_5^2 \vee x_6^2\right) x_7^2 x_8^2\right);$$
$$g_3 = (x_1^2 x_2^2 \vee (x_1^2 \vee x_2^2)(x_3^2 \vee x_4^2) \vee x_3^2 x_4^2) \wedge$$
$$\wedge\, x_5^2 x_6^2 x_7^2 x_8^2.$$

Next, let us construct a GL-model for representing the orphan processors of *Subsystem 2* in the case of its failure. According to the proposed approach, this model is defined as the basic MLE-model of the form $O_2(3, 8) = K(m_2 + 1, n_2) = K(4, 8)$. It is based on a cycle graph with $8 – 4 + 1 = 5$ edges (Fig. 2), and its edge functions have the form:

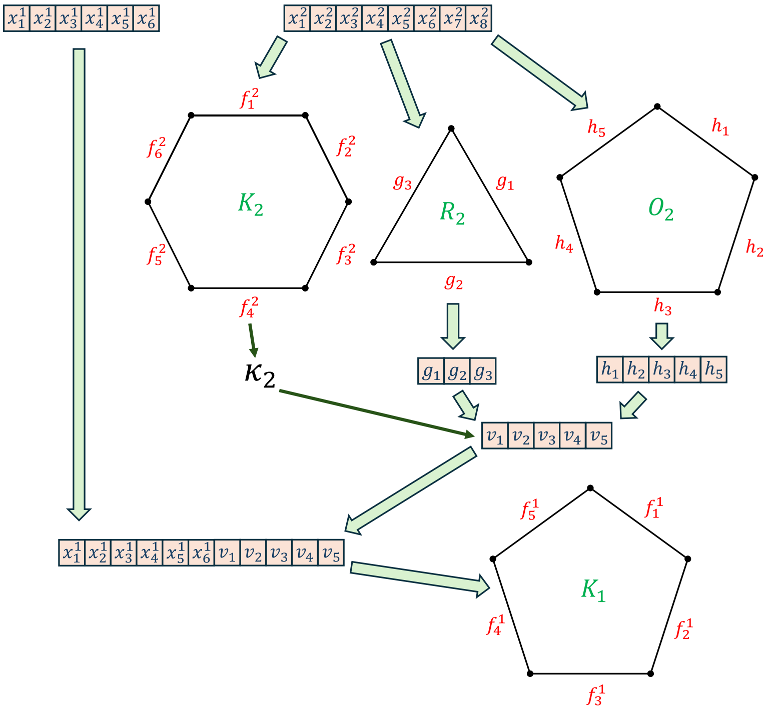

Fig. 2. **GL-model of FTMS from *Example 1***
***Source:* compiled by the authors**

$h_1 = x_1^2 \vee x_2^2 \vee x_3^2 \vee x_4^2;$
$h_2 = (x_1^2 \vee x_2^2 \vee x_3^2 x_4^2)(x_1^2 x_2^2 \vee x_3^2 \vee x_4^2) \vee$
$\vee\, x_5^2 x_6^2 x_7^2 x_8^2;$
$h_3 = (x_1^2 \vee x_2^2)(x_1^2 x_2^2 \vee x_3^2 x_4^2)(x_3^2 \vee x_4^2) \vee$
$\vee \left(x_5^2 \vee x_6^2\right)(x_5^2 x_6^2 \vee x_7^2 x_8^2)(x_7^2 \vee x_8^2);$
$h_4 = x_1^2 x_2^2 x_3^2 x_4^2 \vee$
$\vee \left(x_5^2 \vee x_6^2 \vee x_7^2 x_8^2\right)\left(x_5^2 x_6^2 \vee x_7^2 \vee x_8^2\right);$
$h_5 = x_5^2 \vee x_6^2 \vee x_7^2 \vee x_8^2.$

Let us calculate the values of the components of the state vector of the reserve processors of *Subsystem 2*. Since the numbers of edge functions in the models $R_2(3, 8)$ and $O_2(3, 8)$ are different, let us take $k = \max(k_r, k_o) = \max(3, 5) = 5$ and extend the shorter vector with zero components. Taking into account the note proved above (for $\kappa_2 = 0$ we have $g_i = 0$ for all i), the components of the vector $\mathbf{v} = (v_1, \ldots, v_k)$ are calculated using the simplified formula

$$v_i = g_i \vee \bar{\kappa}_2 h_i, i = 1, \ldots, k.$$

In this example, $k = 5$; therefore:

$v_1 = g_1 \vee \bar{\kappa}_2 h_1;$
$v_2 = g_2 \vee \bar{\kappa}_2 h_2;$
$v_3 = g_3 \vee \bar{\kappa}_2 h_3;$
$v_4 = \bar{\kappa}_2 h_4;$
$v_5 = \bar{\kappa}_2 h_5.$

At the final stage, let us form the extended input vector for the GL-model of *Subsystem 1* [44], which has the form $\mathbf{w} = (x_1^1, \ldots, x_6^1, v_1, \ldots, v_5)$. Since $n_1 = 6$, $m_1 = 2$, and $t = 5$ (the number of components of the vector $\mathbf{v}$), the corresponding basic GL-model is defined as $K_1(m_1 + t, n_1 + t) = K_1(7, 11)$. As the base model, we use the MLE-model. According to (1), it is constructed on a cycle graph with 11 – 7 + 1 = 5 edges (Fig. 2) and has the following edge functions:

$f_1^1 = x_1^1 \vee x_2^1 \vee x_3^1 \vee x_4^1 \vee x_5^1 \vee x_6^1 \vee$
$\vee\, v_1 v_2 v_3 v_4 v_5;$
$f_2^1 = \left(x_1^1 \vee x_2^1 \vee x_3^1 \vee (x_4^1 \vee x_5^1)(x_4^1 x_5^1 \vee x_6^1)\right) \wedge$
$\wedge \left((x_1^1 \vee x_2^1)(x_1^1 x_2^1 \vee x_3^1) \vee x_4^1 \vee x_5^1 \vee x_6^1\right) \vee$
$\vee (v_1 \vee v_2)(v_1 v_2 \vee v_3)(v_1 v_2 v_3 \vee v_4 v_5) \wedge$
$\wedge (v_4 \vee v_5);$
$f_3^1 = (x_1^1 \vee x_2^1 \vee x_3^1 \vee x_4^1 x_5^1 x_6^1) \wedge$
$\wedge ((x_1^1 \vee x_2^1)(x_1^1 x_2^1 \vee x_3^1) \vee$
$\vee (x_4^1 \vee x_5^1)(x_4^1 x_5^1 \vee x_6^1)) \wedge$
$\wedge (x_1^1 x_2^1 x_3^1 \vee x_4^1 \vee x_5^1 \vee x_6^1) \vee (v_1 \vee v_2 \vee v_3) \wedge$
$\wedge \left((v_1 \vee v_2)(v_1 v_2 \vee v_3) \vee v_4 v_5\right) \wedge$
$\wedge (v_1 v_2 v_3 \vee v_4 \vee v_5);$
$f_4^1 = (x_1^1 \vee x_2^1 \vee x_3^1) \wedge$
$\wedge \left((x_1^1 \vee x_2^1)(x_1^1 x_2^1 \vee x_3^1) \vee x_4^1 x_5^1 x_6^1\right) \wedge$
$\wedge \left(x_1^1 x_2^1 x_3^1 \vee (x_4^1 \vee x_5^1)(x_4^1 x_5^1 \vee x_6^1)\right) \wedge$
$\wedge (x_4^1 \vee x_5^1 \vee x_6^1) \vee (v_1 \vee v_2 \vee v_3 \vee v_4 v_5) \wedge$
$\wedge ((v_1 \vee v_2)(v_1 v_2 \vee v_3) \vee v_4 \vee v_5);$
$f_5^1 = (x_1^1 \vee x_2^1)(x_1^1 x_2^1 \vee x_3^1)(x_1^1 x_2^1 x_3^1 \vee x_4^1 x_5^1 x_6^1) \wedge$
$\wedge (x_4^1 \vee x_5^1)(x_4^1 x_5^1 \vee x_6^1) \vee v_1 \vee v_2 \vee v_3 \vee v_4 \vee v_5.$

Thus, the considered example demonstrates the application of the proposed method for constructing a GL-model of a system with a *donor* subsystem and a *recipient* subsystem. The constructed vector $\mathbf{v}$ makes it possible to formally account for the processors of *Subsystem 2* that are available to maintain the operability of *Subsystem 1*, while the extended model $K_1(7, 11)$ integrates these resources into the overall model of its failure behavior.

Example 2. As noted above, the number of reserve processors used to maintain or restore the operability of other subsystems may be additionally limited. Let us consider the pair of subsystems from *Example 1* under the condition that no more than $h = 4$ processors of *Subsystem 2* can participate in restoring the operability of *Subsystem 1* (Fig. 3).

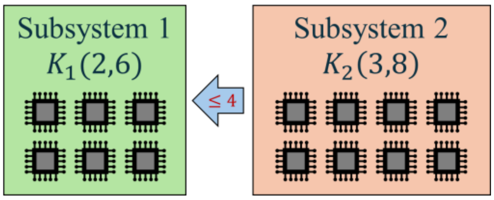

Fig. 3. **FTMS from *Example 2***
Source: **compiled by the authors**

In this case, the GL-models $K_2(3, 8)$, $R_2(3, 8)$, and $O_2(3, 8)$, as well as the formulas for the components of the vector $v_1, \ldots, v_5$, coincide with the corresponding models and formulas presented in *Example 1*. To implement the limitation on the number of reserve processors, let us construct the auxiliary GL-model $L(h, t) = L(4, 5)$, which corresponds to the basic MLE-model $K(t – h + 1, t) = K(5 – 4 + 1, 5) = K(2, 5)$. According to (1), this model is constructed on a cycle graph with 5 – 2 + 1 = 4 edges (Fig. 4) and has the following edge functions:

$e_1 = v_1 \vee v_2;$
$e_2 = v_1 v_2 \vee v_3;$
$e_3 = v_1 v_2 v_3 \vee v_4 v_5;$
$e_4 = v_4 \vee v_5.$

Then the extended input vector for the GL-model of *Subsystem 1* has the form $\mathbf{w} = (x_1^1, \ldots, x_6^1, e_1, \ldots, e_4)$. Since $n_1 = 6$, $m_1 = 2$, and $h = 4$, the corresponding MLE-model is defined as $K_1(m_1 + h, n_1 + h) = K_1(6, 10)$. According to (1), this model is constructed on a cycle graph with 10 – 6 + 1 = 5 edges (Fig. 4) and has the following edge functions:

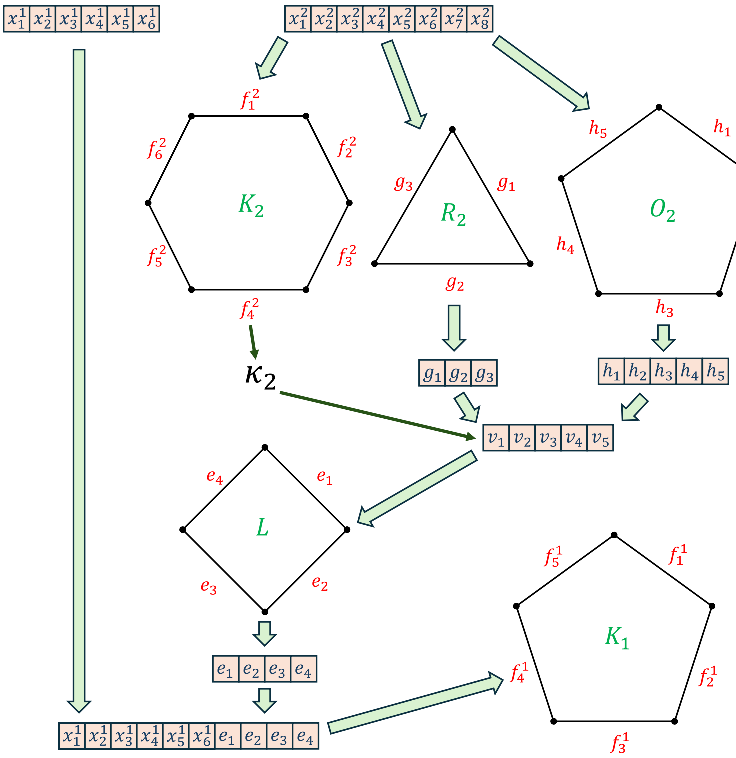

Fig. 4. **GL-model of FTMS from *Example 2***
***Source:* compiled by the authors**

$$f_1^1 = x_1^1 \vee x_2^1 \vee x_3^1 \vee x_4^1 \vee x_5^1 \vee x_6^1 e_1 e_2 e_3 e_4;$$
$$f_2^1 = (x_1^1 \vee x_2^1 \vee x_3^1 \vee x_4^1 x_5^1) \wedge$$
$$\wedge \left((x_1^1 \vee x_2^1)(x_1^1 x_2^1 \vee x_3^1) \vee x_4^1 \vee x_5^1 \right) \vee$$
$$\vee (x_6^1 \vee e_1)(x_6^1 e_1 \vee e_2)(x_6^1 e_1 e_2 \vee e_3 e_4)(e_3 \vee e_4);$$
$$f_3^1 = (x_1^1 \vee x_2^1 \vee x_3^1) \wedge$$
$$\wedge \left((x_1^1 \vee x_2^1)(x_1^1 x_2^1 \vee x_3^1) \vee x_4^1 x_5^1 \right) \wedge$$
$$\wedge (x_1^1 x_2^1 x_3^1 \vee x_4^1 \vee x_5^1) \vee (x_6^1 \vee e_1 \vee e_2) \wedge$$
$$\wedge \left((x_6^1 \vee e_1)(x_6^1 e_1 \vee e_2) \vee e_3 e_4 \right) \wedge$$
$$\wedge (x_6^1 e_1 e_2 \vee e_3 \vee e_4);$$
$$f_4^1 = (x_1^1 \vee x_2^1)(x_1^1 x_2^1 \vee x_3^1)(x_1^1 x_2^1 x_3^1 \vee x_4^1 x_5^1) \wedge$$
$$\wedge (x_4^1 \vee x_5^1) \vee (x_6^1 \vee e_1 \vee e_2 \vee e_3 e_4) \wedge$$
$$\wedge \left((x_6^1 \vee e_1)(x_6^1 e_1 \vee e_2) \vee e_3 \vee e_4 \right);$$
$$f_5^1 = x_1^1 x_2^1 x_3^1 x_4^1 x_5^1 \vee x_6^1 \vee e_1 \vee e_2 \vee e_3 \vee e_4.$$

Thus, the example illustrates the use of the auxiliary GL-model *L*(*h*, *t*) to introduce the limitation *h* and the corresponding construction of the extended GL-model of the *recipient* subsystem.

Example 3. Consider a system consisting of four subsystems (Fig. 5). *Subsystem 1* contains $n_1 = 8$ processors and is tolerant to the failure of no more than $m_1 = 3$ of them. *Subsystem 2* contains $n_2 = 7$ processors and is tolerant to the failure of no more than $m_2 = 2$ of them. *Subsystem 3* contains $n_3 = 7$ processors and is tolerant to the failure of no more than $m_3 = 3$ of them. *Subsystem 4* contains $n_4 = 5$ processors and is tolerant to the failure of no more than $m_4 = 2$ of them.

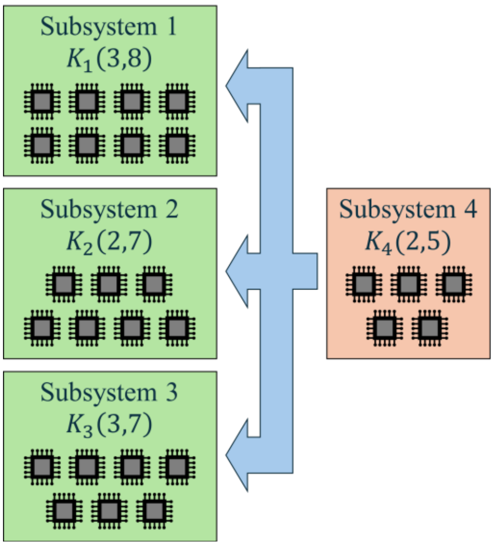

Fig. 5. **FTMS from *Example 3***
***Source:* compiled by the authors**

It is assumed that *Subsystem 4* acts as the *donor* for *Subsystems 1-3*: when it remains operable, its *redundant* processors may be used, and in the case of its failure, its operational *orphan* processors may be used. The use of *donor* resources is organized according to the principle of sliding redundancy. It is required to construct a GL-model that represents the failure behavior of *Subsystems 1-3* taking into account the sliding redundancy of *Subsystem 4*.

The elements of the system state vector corresponding to the processors of *Subsystem 1* will be denoted as $x_1^1, \ldots, x_8^1$; those corresponding to the processors of *Subsystem 2* as $x_1^2, \ldots, x_7^2$; those corresponding to the processors of *Subsystem 3* as $x_1^3, \ldots, x_7^3$; and those corresponding to the processors of *Subsystem 4* as $x_1^4, \ldots, x_5^4$.

First, let us construct the MLE-model $K_4(2, 5)$ for *Subsystem 4*. According to (1), it is based on a cycle graph with 5 – 2 + 1 = 4 edges (Fig. 6) and has the following edge functions:

$f_1^4 = x_1^4 \vee x_2^4;$
$f_2^4 = x_1^4 x_2^4 \vee x_3^4;$
$f_3^4 = x_1^4 x_2^4 x_3^4 \vee x_4^4 x_5^4;$
$f_4^4 = x_4^4 \vee x_5^4.$

The Boolean result (the state of *Subsystem 4*) determined using the model $K_4(2, 5)$ will be denoted as $\kappa_4 \in \{0,1\}$.

Next, let us construct the model $R_4(2, 5)$ for representing the redundant processors of *Subsystem 4* in the case where it remains operable. According to the described method, this model is defined as the dual of the MLE-model $K(4, 5)$, that is, $R_4(2, 5) = K^*(4, 5)$. The model is based on a cycle graph with 5 – 4 + 1 = 2 edges (Fig. 6) and has the following edge functions:

$g_1 = x_1^4 x_2^4 x_3^4 (x_4^4 \vee x_5^4);$
$g_2 = (x_1^4 x_2^4 \vee (x_1^4 \vee x_2^4) x_3^4) x_4^4 x_5^4.$

Let us also construct the model $O_4(2, 5)$ for representing the operational orphan processors of *Subsystem 4* in the case of its failure. According to the described approach, this model is defined as the basic MLE-model of the form $K(3, 5)$. It is based on a cycle graph with 5 – 3 + 1 = 3 edges (Fig. 6) and has the following edge functions:

$h_1 = x_1^4 \vee x_2^4 \vee x_3^4;$
$h_2 = (x_1^4 \vee x_2^4)(x_1^4 x_2^4 \vee x_3^4) \vee x_4^4 x_5^4;$
$h_3 = x_1^4 x_2^4 x_3^4 \vee x_4^4 \vee x_5^4.$

Now let us determine the values of the components of the state vector of the sliding redundancy processors $\mathbf{v} = (v_1, v_2, v_3)$, taking into account that $k = \max(k_r, k_o) = \max(2, 3) = 3$. Since $k_r = 2 < k$, we set $g_3 = 0$. We obtain:

$v_1 = g_1 \vee \bar{\kappa}_4 h_1;$
$v_2 = g_2 \vee \bar{\kappa}_4 h_2;$
$v_3 = \bar{\kappa}_4 h_3.$

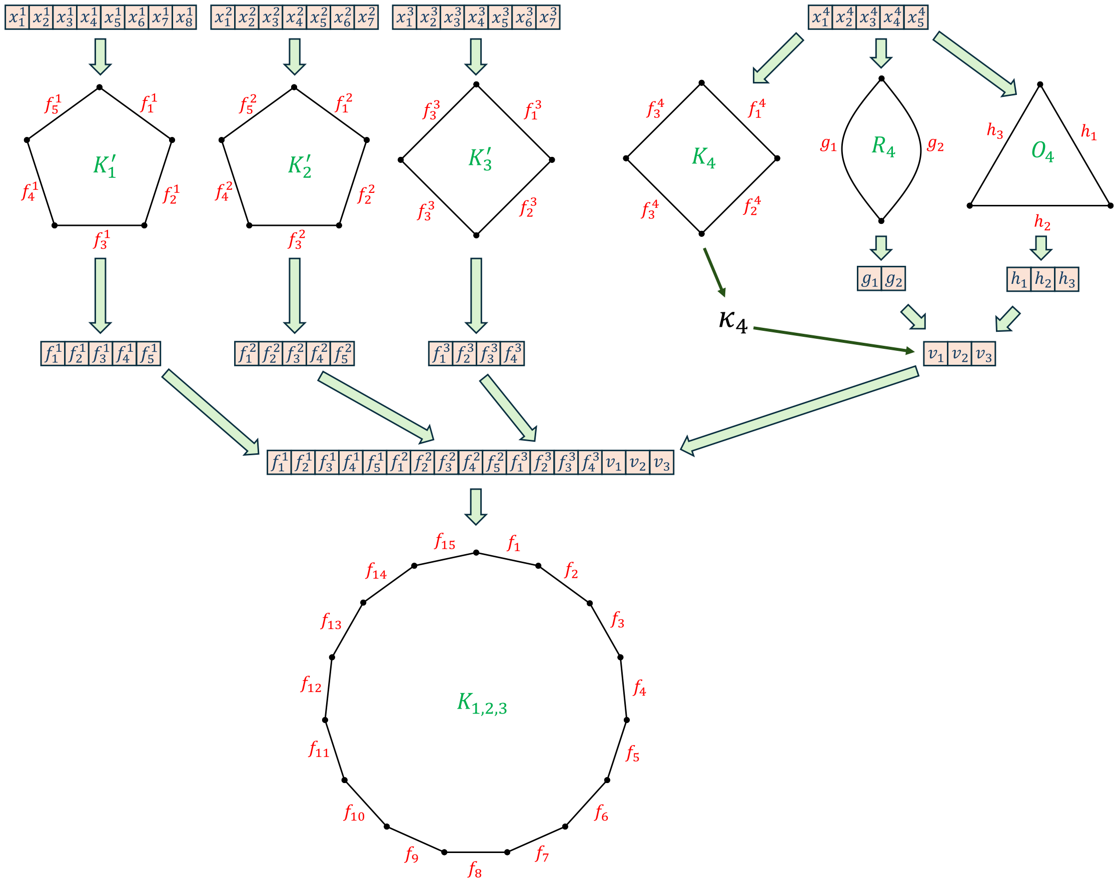

Fig. 6. **GL-model of FTMS from *Example 3***
***Source:* compiled by the authors**

Next for the recipient subsystems (*Subsystems 1-3*) let us construct auxiliary MLE-models of the form $K_i'(m_i+1, n_i)$. The values of their edge functions will be used to form Boolean vectors characterizing the need of each subsystem for failure compensation through sliding redundancy.

Subsystem 1. Since $n_1 = 8$ and $m_1 = 3$, the corresponding auxiliary MLE-model has the form $K_1'(4,8)$. According to (1), it is based on a cycle graph with 8 – 4 + 1 = 5 edges (Fig. 6) and has the following edge functions:

$$f_1^1 = x_1^1 \vee x_2^1 \vee x_3^1 \vee x_4^1;$$
$$f_2^1 = (x_1^1 \vee x_2^1 \vee x_3^1 x_4^1)(x_1^1 x_2^1 \vee x_3^1 \vee x_4^1) \vee$$
$$\vee\, x_5^1 x_6^1 x_7^1 x_8^1;$$
$$f_3^1 = (x_1^1 \vee x_2^1)(x_1^1 x_2^1 \vee x_3^1 x_4^1)(x_3^1 \vee x_4^1) \vee$$
$$\vee\, (x_5^1 \vee x_6^1)(x_5^1 x_6^1 \vee x_7^1 x_8^1)(x_7^1 \vee x_8^1);$$
$$f_4^1 = x_1^1 x_2^1 x_3^1 x_4^1 \vee$$
$$\vee\, (x_5^1 \vee x_6^1 \vee x_7^1 x_8^1)(x_5^1 x_6^1 \vee x_7^1 \vee x_8^1);$$
$$f_5^1 = x_5^1 \vee x_6^1 \vee x_7^1 \vee x_8^1.$$

Subsystem 2. Since $n_2 = 7$ and $m_2 = 2$, the corresponding auxiliary MLE-model has the form $K_2'(3,7)$. It is based on a cycle graph with 7 – 3 + 1 = 5 edges (Fig. 6) and has the following edge functions:

$$f_1^2 = x_1^2 \vee x_2^2 \vee x_3^2 x_4^2;$$
$$f_2^2 = x_1^2 x_2^2 \vee x_3^2 \vee x_4^2;$$
$$f_3^2 = (x_1^2 \vee x_2^2)(x_1^2 x_2^2 \vee x_3^2 x_4^2)(x_3^2 \vee x_4^2) \vee x_5^2 x_6^2 x_7^2;$$
$$f_4^2 = x_1^2 x_2^2 x_3^2 x_4^2 \vee \left(x_5^2 \vee x_6^2\right)\left(x_5^2 x_6^2 \vee x_7^2\right);$$
$$f_5^2 = x_5^2 \vee x_6^2 \vee x_7^2.$$

Subsystem 3. Since $n_3 = 7$ and $m_3 = 3$, the corresponding auxiliary MLE-model has the form $K_3'(4,7)$. It is based on a cycle graph with 7 – 4 + 1 = 4 edges (Fig. 6) and has the following edge functions:

$$f_1^3 = x_1^3 \vee x_2^3 \vee x_3^3 \vee x_4^3;$$
$$f_2^3 = (x_1^3 \vee x_2^3 \vee x_3^3 x_4^3)(x_1^3 x_2^3 \vee x_3^3 \vee x_4^3) \vee$$
$$\vee\, x_5^3 x_6^3 x_7^3;$$
$$f_3^3 = (x_1^3 \vee x_2^3)(x_1^3 x_2^3 \vee x_3^3 x_4^3)(x_3^3 \vee x_4^3) \vee$$
$$\vee \left(x_5^3 \vee x_6^3\right)\left(x_5^3 x_6^3 \vee x_7^3\right);$$
$$f_4^3 = x_1^3 x_2^3 x_3^3 x_4^3 \vee x_5^3 \vee x_6^3 \vee x_7^3.$$

Let us form the Boolean vector $\mathbf{w} = (f_1^1, \dots, f_5^1, f_1^2, \dots, f_5^2, f_1^3, \dots, f_4^3, v_1, v_2, v_3)$, which consists of the values of the edge functions of the auxiliary models $K_1'(4,8)$, $K_2'(3,7)$, $K_3'(4,7)$, and the components of the vector **v**, which represents the states of the sliding redundancy processors.

Next, let us construct the basic MLE-model $K(3, 17)$, which corresponds to the condition of simultaneous operability of *Subsystems 1-3* taking into account the sliding redundancy of *Subsystem 4*. According to (1), this model is based on a cycle graph with 17 – 3 + 1 = 15 edges (Fig. 6) and has the following edge functions:

$$f_1 = f_1^1 \vee f_2^1 \vee f_3^1;$$
$$f_2 = (f_1^1 \vee f_2^1)(f_1^1 f_2^1 \vee f_3^1) \vee f_4^1 f_5^1;$$
$$f_3 = f_1^1 f_2^1 f_3^1 \vee f_4^1 \vee f_5^1;$$
$$f_4 = (f_1^1 \vee f_2^1)(f_1^1 f_2^1 \vee f_3^1)(f_1^1 f_2^1 f_3^1 \vee f_4^1 f_5^1) \wedge$$
$$\wedge\, (f_4^1 \vee f_5^1) \vee f_1^2 f_2^2 f_3^2 f_4^2;$$
$$f_5 = f_1^1 f_2^1 f_3^1 f_4^1 f_5^1 \vee (f_1^2 \vee f_2^2)(f_1^2 f_2^2 \vee f_3^2 f_4^2) \wedge$$
$$\wedge\, (f_3^2 \vee f_4^2);$$
$$f_6 = f_1^2 \vee f_2^2 \vee f_3^2 f_4^2;$$
$$f_7 = f_1^2 f_2^2 \vee f_3^2 \vee f_4^2;$$
$$f_8 = (f_1^1 \vee f_2^1)(f_1^1 f_2^1 \vee f_3^1)(f_1^1 f_2^1 f_3^1 \vee f_4^1 f_5^1) \wedge$$
$$\wedge\, (f_4^1 \vee f_5^1)(f_1^1 f_2^1 f_3^1 f_4^1 f_5^1 \vee f_1^2 f_2^2 f_3^2 f_4^2)(f_1^2 \vee f_2^2) \wedge$$
$$\wedge\, (f_1^2 f_2^2 \vee f_3^2 f_4^2)(f_3^2 \vee f_4^2) \vee f_5^2 f_1^3 f_2^3 f_3^3 f_4^3 v_1 v_2 v_3;$$
$$f_9 = f_1^1 f_2^1 f_3^1 f_4^1 f_5^1 f_1^2 f_2^2 f_3^2 f_4^2 \vee \left(f_5^2 \vee f_1^3\right) \wedge$$
$$\wedge \left(f_5^2 f_1^3 \vee f_2^3 f_3^3\right)(f_2^3 \vee f_3^3) \wedge$$
$$\wedge \left(f_5^2 f_1^3 f_2^3 f_3^3 \vee f_4^3 v_1 v_2 v_3\right)(f_4^3 \vee v_1) \wedge$$
$$\wedge\, (f_4^3 v_1 \vee v_2 v_3)(v_2 \vee v_3);$$
$$f_{10} = f_5^2 \vee f_1^3 \vee f_2^3 f_3^3;$$
$$f_{11} = f_5^2 f_1^3 \vee f_2^3 \vee f_3^3;$$
$$f_{12} = \left(f_5^2 \vee f_1^3\right)\left(f_5^2 f_1^3 \vee f_2^3 f_3^3\right)(f_2^3 \vee f_3^3) \vee$$
$$\vee\, f_4^3 v_1 v_2 v_3;$$
$$f_{13} = f_5^2 f_1^3 f_2^3 f_3^3 \vee$$
$$\vee\, (f_4^3 \vee v_1)(f_4^3 v_1 \vee v_2 v_3)(v_2 \vee v_3);$$
$$f_{14} = f_4^3 \vee v_1 \vee v_2 v_3;$$
$$f_{15} = f_4^3 v_1 \vee v_2 \vee v_3.$$

Thus, the example illustrates the construction of a GL-model for a system with multiple *recipient* subsystems and sliding redundancy, where the state of the *donor* subsystem and the need of the subsystems for failure compensation are integrated into a single model $K(3, 17)$.

DISCUSSION OF RESULTS

To validate the correctness of the proposed method, a series of computational experiments were conducted using the GL-models constructed in the presented examples, as well as other models obtained for systems with different parameters (including variations in the number of processors, fault-tolerance levels, and configurations of inter-subsystem redundancy).

For each generated state vector, the output of the corresponding GL-model (i.e., the connectivity of the model graph) was compared with the expected operability state of the system, determined independently based on its formal definition. This comparison made it possible to verify the correctness of the constructed models over a wide range of input conditions.

The experiments were implemented programmatically, allowing automated evaluation over large sets of state vectors, including both randomly generated vectors and, in some cases, exhaustive enumeration of all possible states. The obtained results confirmed that the GL-models constructed using the proposed method correctly represent the failure behavior of the considered systems.

As an illustration of the described experimental procedure, let us consider the system from *Example 1*. Suppose that the 2nd, 3rd, and 5th processors of *Subsystem 1*, as well as the 2nd and 7th processors of *Subsystem 2*, have failed. Then the system state vector has the form **x** = 10010110111101.

In this situation, since $m_1 = 2$, *Subsystem 1* lacks one processor to restore operability. At the same time, since $m_2 = 3$, *Subsystem 2* remains operable and contains one *redundant* processor. Thus, *Subsystem 2* can act as a *donor* by providing the resources of one of its processors to *Subsystem 1*, which acts as the *recipient* subsystem, and this is sufficient to restore its operability.

Now let us consider the behavior of the constructed GL-models for this vector. All edge functions of the model $K_2(3, 8)$ take the value one; therefore, $\kappa_2 = 1$. Among the edge functions of the model $R_2(3, 8)$, only the function g_2 takes the value one. At the same time, in the model $O_2(3, 8)$ all edge functions take the value one.

Thus, **v** = 01000. The corresponding extended vector has the form **w** = 10010101000. For this vector, in the model $K_1(7, 11)$ all edge functions except f_4^1 take the value one. At the same time, the graph of the model remains connected, which corresponds to the operable state of *Subsystem 1*.

If the system state vector has the form **x** = 01100000101100, that is, only the 2nd and 3rd processors of *Subsystem 1* and the 3rd, 5th, and 6th processors of *Subsystem 2* remain operational, then *Subsystem 1* requires two additional processors to restore operability. At the same time, *Subsystem 2* is failed, but it contains three *orphan* processors that can be used to restore the operability of *Subsystem 1*.

For the specified vector **x**, three edge functions of the model $K_2(3, 8)$ take the value zero, namely f_1^2, f_3^2, and f_4^2. Accordingly, $\kappa_2 = 0$. All edge functions of the model $R_2(3, 8)$ also take the value zero, which is consistent with the statement proved above. In the model $O_2(3, 8)$, two edge functions – h_2 and h_3 – take the value zero.

Thus, **v** = 10011, and **w** = 01100010011. For this vector, in the model $K_1(7, 11)$ all edge functions take the value one, and the graph of this model remains connected, which, as expected, corresponds to the operable state of *Subsystem 1*.

Finally, let us consider the case where in *Subsystem 1* the 1st, 2nd, 4th, and 6th processors are failed, and in *Subsystem 2* the 2nd and 8th processors are failed; that is, the system state vector has the form **x** = 00101010111110. In this case, *Subsystem 1* requires two processors to restore operability, whereas *Subsystem 2* remains operable and contains only one *redundant* processor, which is insufficient to restore the operability of *Subsystem 1*.

For this vector **x**, all edge functions of the model $K_2(3, 8)$ take the value one; its graph remains connected, and therefore $\kappa_2 = 1$. Among the edge functions of the model $R_2(3, 8)$, only the function g_2 takes the value one. At the same time, all edge functions of the model $O_2(3, 8)$ also take the value one.

Thus, **v** = 01000, and **w** = 00101001000. For this vector, two edge functions of the model $K_1(7, 11)$, namely f_3^1 and f_4^1, take the value zero. As a result, the cycle graph of this model loses connectivity, which corresponds to the failure of *Subsystem 1*.

The considered scenarios show that the constructed GL-models correctly represent the behavior of the system under different combinations of processor failures. In particular, the models adequately account for the use of both the *redundant* processors of the *donor* subsystem when it remains operable and the *orphan* processors when it fails, as well as situations in which the available resources are insufficient to compensate for the failures of the *recipient* subsystem. The obtained results are consistent with the expected logic of system operation and confirm the applicability of the proposed approach for modeling fault-tolerant multiprocessor systems with inter-subsystem redundancy. In addition, the results obtained using the GL-models correspond to analytical estimates of the number of failures that can be compensated by the resources of the *donor* subsystem.

It is also advisable to compare the complexity of the expressions of the edge functions of the GL-models obtained using the proposed method with those of GL-models constructed by known methods. As noted above, previously developed methods did not cover cases of fault-tolerant multiprocessor systems with such complex behavior. Therefore, for comparison, we use a certain basic GL-model, without considering that, according to known methods, it would subsequently require additional modification. Such modifications typically increase the complexity of the corresponding model.

Thus, in *Example 1* and *Example 2*, the two subsystems together contain $n = n_1 + n_2 = 6 + 8 = 14$ processors and, in the best case, are tolerant to the failure of $m = m_1 + m_2 = 2 + 3 = 5$ of them. Therefore, the MLE-model $K(m, n) = K(5, 14)$ can be considered as the basic model.

Table 1 presents a comparison of the complexity of the expressions of the edge functions of the GL-models constructed in *Example 1* and *Example 2*, as well as the basic MLE-model $K(5, 14)$. As can be seen from the table, the complexity of the GL-models constructed using the method proposed in this work is at least comparable with the complexity of the corresponding basic MLE-model (which, according to previously known approaches, would still require further modification). This indicates that the proposed method makes it possible to describe complex schemes of inter-subsystem redundancy without a significant increase in the complexity of the GL-model.

Table 1. **Number of logical operations in the edge-function expressions of the GL-models for *Examples 1, 2* and basic GL-model $K(5, 14)$**

Model	Disj.	Conj.	Inv.	Binary ops.	Total ops.
Example 1	124	124	5	248	253
Example 2	109	114	5	223	228
Basic $K(5, 14)$	110	128	0	238	238

Source: **compiled by the authors**

CONCLUSIONS

The paper proposes a generalized method and algorithm for constructing GL-models for fault-tolerant multiprocessor systems containing *donor* and *recipient* subsystems. Unlike previously proposed methods, this approach takes into account the possibility of using the processors of the *donor* subsystem both when it remains operable and when it fails.

The proposed approach can be used as a basis for constructing GL-models for fault-tolerant multiprocessor systems with complex behavior and can be combined with other known approaches to constructing GL-models. In particular, the possibility of combining it with the approach that allows modeling fault-tolerant multiprocessor systems employing sliding redundancy is demonstrated.

Experiments have been conducted and examples have been presented that further confirm the correctness of the GL-models constructed using the proposed method. It is also shown that the complexity of the expressions of the edge functions of these models is comparable to the complexity of the expressions of the edge functions of the corresponding basic model, which would subsequently require additional modification.

The paper considers subsystems whose behavior corresponds to that of basic systems. At the same time, the proposed approach opens up possibilities for modeling subsystems with more complex failure behavior. This constitutes one of the promising directions for further research.

Another promising direction for further research is the search for ways to optimize the computation of the edge functions of the models. In particular, this can be achieved by reusing the values of parts of the expressions of the edge functions across different auxiliary models.

REFERENCES

1. Khilenko, V. V., Denysiuk, S. P., Bielokha, H. S., Stepanov, O. V. & Kang, L.-W. “Optimizing the control of a transactive energy system with periodically changing architecture based on internet of energy”. *Herald of Advanced Information Technology*. 2026; 9 (1): 85–99, https://www.scopus.com/pages/publications/105031926170. DOI: https://doi.org/10.15276/hait.09.2026.07.

2. Nazarova, O. S., Osadchyy, V. V. & Rudim, B. Y. “Computer simulation of the microprocessor liquid level automatic control system”. *Applied Aspects of Information Technology*. 2023; 6 (2): 163–174. DOI: https://doi.org/10.15276/aait.06.2023.12.

3. Kotov, D. O. “A generalized model of an adaptive information-control system of a car with multi-sensor channels of information interaction”. *Applied Aspects of Information Technology*. 2022; 5 (1): 25–34. DOI: https://doi.org/10.15276/aait.05.2022.2.

4. Shavarskyi, M. A. & Kryvechuk, Y. P. “Anomaly detection on time series data for autonomous robot operation”. *Herald of Advanced Information Technology.* 2026; 9 (1): 46–59, https://www.scopus.com/pages/publications/105031923827. DOI: https://doi.org/10.15276/hait.09.2026.04.

5. Kovalev, I. S., Drozd, O. V., Rucinski, A., Drozd, M. O., Antoniuk, V. V. & Sulima Y. Y. “Development of computer system components in critical applications: Problems, their origins and solutions”. *Herald of Advanced Information Technology.* 2020; 3 (4): 252–262. DOI: https://doi.org/10.15276/hait.04.2020.4.

6. Antoniuk, V. V., Drozd, M. O. & Drozd, O. B. “Power-oriented checkability and monitoring of the current consumption in FPGA projects of the critical applications”. *Applied Aspects of Information Technology*. 2019; 2 (2): 105–114. DOI: https://doi.org/10.15276/aait.02.2019.2.

7. Myrnneko, M. D. & Shevel, V. V. “Technological platforms, tools, and standardization in the unified information space for automated Unmanned Aerial Vehicle design”. *Applied Aspects of Information Technology*. 2025; 8 (3): 316–333. DOI: https://doi.org/10.15276/aait.08.2025.21.

8. Joshi, H. & Sinha, N. K. “Adaptive fault tolerant control design for stratospheric airship with actuator faults”. *IFAC-PapersOnLine*. 2022; 55 (1): 819–825, https://www.scopus.com/pages/publications/85132157930. DOI: https://doi.org/10.1016/j.ifacol.2022.04.134.

9. Abbaspour, A., Mokhtari, S., Sargolzaei, A. & Yen, K. K. “A survey on active fault-tolerant control systems”. *Electronics*. 2020; 9 (9): 1–23, https://www.scopus.com/pages/publications/85090902832. DOI: https://doi.org/10.3390/electronics9091513.

10. Nedeljkovic, J. N., Dosic, S. M. & Nikolic, G. S. “A survey of hardware fault tolerance techniques”. *2023 58th International Scientific Conference on Information, Communication and Energy Systems and Technologies, ICEST. Proceedings*. 2023. p. 223–226, https://www.scopus.com/pages/publications/85167870342. DOI: https://doi.org/10.1109/ICEST58410.2023.10187275.

11. Billinton, R. & Allan, R. N. “Reliability evaluation of engineering systems. Concepts and techniques”. *Springer New York*. 1992. DOI: https://doi.org/10.1007/978-1-4899-0685-4.

12. Yi, H., Cui, L. & Gao, H. “Reliabilities of some multistate Consecutive-k systems”. *IEEE Transactions on Reliability*. 2020; 69 (2): 414–429. DOI: https://doi.org/10.1109/TR.2019.2897726.

13. Jianu, M., Daus, L., Dragoi, V. F. & Beiu, V. “Reliability polynomials of consecutive-k-out-of-n:F systems have unbounded roots”. *Networks*. 2023; 82 (3): 222–228, https://www.scopus.com/pages/publications/85163222383. DOI: https://doi.org/10.1002/net.22168.

14. Hu, B. & Seiler, P. “Pivotal decomposition for reliability analysis of fault tolerant control systems on unmanned aerial vehicles”. *Reliability Engineering & System Safety*. 2015; 140: 130–141, https://www.scopus.com/pages/publications/84928741612. DOI: https://doi.org/10.1016/j.ress.2015.04.005.

15. Dong, H., Xie, H. & Xu, H. “Reliability evaluation of motor controllers based on FMEA and DFTA”. *Hangkong Dongli Xuebao/Journal of Aerospace Power*. 2025; 40 (9): 20230656, https://www.scopus.com/pages/publications/105011205376.
DOI: https://doi.org/10.13224/j.cnki.jasp.20230656.

16. Dagal, I. “Probabilistic fault tree analysis and dynamic redundancy optimization for next-generation avionic flight control systems”. *Reliability Engineering and System Safety*. 2026; 266: 111841, https://www.scopus.com/pages/publications/105020942559. DOI: https://doi.org/10.1016/j.ress.2025.111841.

17. Wu, J., Chen, C., Ma, Y., Xiu, Z., Cheng, Z., Pan, Y. & Song, S. “Integrated fault tree and case analysis for equipment conventional fault IETM diagnosis”. *Sensors*. 2025; 25 (17): 5231, https://www.scopus.com/pages/publications/105015822431. DOI: https://doi.org/10.3390/s25175231.

18. Romankevich, A., Feseniuk, A., Maidaniuk, I. & Romankevich, V. “Fault-tolerant multiprocessor systems reliability estimation using statistical experiments with GL-models”. *Advances in Intelligent Systems and Computing*. 2019; 754: 186–193, https://www.scopus.com/pages/publications/85047465084. DOI: https://doi.org/10.1007/978-3-319-91008-6_19.

19. Dembińska, A. & Nikolov, N. I. “Reliability properties of k-out-of-n systems with several cold standby units”. *Journal of Computational and Applied Mathematics*. 2026; 477: 117203, https://www.scopus.com/pages/publications/105021472679.
DOI: https://doi.org/10.1016/j.cam.2025.117203.

20. Mannai, N. & Jarray, M. “Optimal design of k-out-of-n:G system under load redistribution and stochastic mission duration”. *Reliability Engineering and System Safety*. 2026; 269: 112029, https://www.scopus.com/pages/publications/105023957393.
DOI: https://doi.org/10.1016/j.ress.2025.112029.

21. Yin, J. & Cui, L. “Reliability for consecutive-k-out-of-n:F systems with shared components between adjacent subsystems”. *Reliability Engineering & System Safety*. 2021; 210: 107532, https://www.scopus.com/pages/publications/85100736521. DOI: https://doi.org/10.1016/j.ress.2021.107532.

22. Chopra, G. & Ram, M. “Linear Consecutive-k-out-of-n:G System Reliability Analysis”. *Journal of Reliability and Statistical Studies*. 2022; 15 (2): 669–692. DOI: https://doi.org/10.13052/jrss0974-8024.15211.

23. Yin, J., Balakrishnan, N. & Cui, L. “Efficient reliability computation of consecutive- k-out-of-n: F systems with shared components”. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. 2024; 238 (1): 122–135, https://www.scopus.com/pages/publications/85142023916. DOI: https://doi.org/10.1177/1748006X221130540.

24. Eryilmaz, S. & Kan, C. “The mean number of failed components in discrete time consecutive k-out-of-n: F system and its application to parameter estimation and optimal age-based preventive replacement”. *Reliability Engineering and System* Safety. 2025; 263: 111229, https://www.scopus.com/pages/publications/105005869605.
DOI: https://doi.org/10.1016/j.ress.2025.111229.

25. Triantafyllou, I. S. “Combined m-Consecutive-k-Out-of-n: F and Consecutive k_c-Out-of-n:F structures with cold standby redundancy”. *Mathematics*. 2023; 11 (12): 1–13, https://www.scopus.com/pages/publications/85164204439. DOI: https://doi.org/10.3390/math11122597.

26. Triantafyllou, I. S. “On the combined m-consecutive-k-out-of-n: F and consecutive k_c-out-of-n: F reliability system: Some advances”. *Reliability Modeling in Industry 4.0*. 2023. 463–476, https://www.scopus.com/pages/publications/85161955494. DOI: https://doi.org/10.1016/B978-0-323-99204-6.00004-2.

27. Lin, C., Cui L. R., Coit D. W. & Lv, M. “Reliability modeling on consecutive-k_r-out-of-n_r:F linear zigzag structure and circular polygon structure”. *IEEE Transactions on Reliability*. 2016; 65 (3): 1509–1521, https://www.scopus.com/pages/publications/84973885922. DOI: https://doi.org/10.1109/TR.2016.2570545.

28. Yin, J., Cui, L. & Balakrishnan N. “Reliability of consecutive-(k,l)-out-of-n: F systems with shared components under non-homogeneous Markov dependence”. *Reliability Engineering & System Safety*. 2022; 224: 108549, https://www.scopus.com/pages/publications/85129557917.
DOI: https://doi.org/10.1016/j.ress.2022.108549.

29. Nakamura, T., Homma, I. & Yamamoto, H. “Birnbaum importance-based simulated annealing algorithm for solving the component assignment problem of linear connected-(r, s)-out-of-(m, n):F lattice systems”. *International Journal of System Assurance Engineering and Management*. 2024; 15 (4): 1407–1414, https://www.scopus.com/pages/publications/85144872424.
DOI: https://doi.org/10.1007/s13198-022-01848-2.

30. Nakamura, T., Yamamoto, H. & Akiba, T. “Reliability of a toroidal connected-(r,s)-out-of-(m,n):F lattice system”. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. 2022; 236 (2): 329–338, https://www.scopus.com/pages/publications/85077453315. DOI: https://doi.org/10.1177/1748006X19893533.

31. Lu, J., Yi, H., Li, X. & Balakrishnan, N. “Joint reliability of two consecutive-(1, l) or (2, k)-out-of-(2, n): F type systems and its application in smart street light deployment”. *Methodology and Computing in Applied Probability*. 2023; 25 (1): 33, https://www.scopus.com/pages/publications/85148634528. DOI: https://doi.org/10.1007/s11009-023-09984-3.

32. Amirian, Y., Khodadadi, A. & Chatrabgoun, O. “Exact reliability for a consecutive circular k-out-of-r-from-n:F system with equal and unequal component probabilities”. *International Journal of Reliability, Quality and Safety Engineering*. 2020; 27 (1), https://www.scopus.com/pages/publications/85069848232. DOI: https://doi.org/10.1142/S0218539320500035.

33. Chachra, A., Ram, M. & Kumar, A. “A pythagorean fuzzy approach to consecutive k-out-of-r-from-n system reliability modelling”. *International Journal of System Assurance Engineering and Management*.

2024, https://www.scopus.com/pages/publications/85200035598. DOI: https://doi.org/10.1007/s13198-024-02435-3.

34. Zhu, X., Boushaba, M. & Reghioua, M. "Reliability and joint reliability importance in a Consecutive-k-Within-m-out-of-n:F system with Markov-Dependent cmponents". *IEEE Transactions on Reliability*. 2016; 65 (2): 802–815, https://www.scopus.com/pages/publications/84943426081. DOI: https://doi.org/10.1109/TR.2015.2484079.

35. Torrado, N. "Tail behaviour of consecutive 2-within-m-out-of-n systems with nonidentical components". *Applied Mathematical Modelling*. 2015; 39 (15): 4586–4592, https://www.scopus.com/pages/publications/84937631892. DOI: https://doi.org/10.1016/j.apm.2014.12.042.

36. Triantafyllou, I. "m-Consecutive-k-out-of-n: F Structures with a single change point". *Mathematics*. 2020; 8 (12): 2203, https://www.scopus.com/pages/publications/85097534209. DOI: https://doi.org/10.3390/math8122203.

37. Nashwan, I. "Reliability and Failure Probability Functions of the m-Consecutive-k-out-of-n: F Linear and Circular Systems". *Baghdad Science Journal*. 2021; 18 (2): 430, https://www.scopus.com/pages/publications/85100161580.
DOI: https://doi.org/10.21123/bsj.2021.18.2.0430.

38. Zhu, X., Boushaba, M., Coit, D. W. & Benyahia, A. "Reliability and importance measures for m-consecutive-k,l-out-of-n system with non-homogeneous Markov-dependent components". *Reliability Engineering and System Safety, Elsevier*. 2017; 167 (C): 1–9, https://www.scopus.com/pages/publications/85019236438. DOI: https://doi.org/10.1016/j.ress.2017.05.023.

39. Özbey, F. "Reliability evaluation of m-consecutive-k,l-out-of-n: F system subjected to shocks". *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. 2022; 236 (6): 1135–1146, https://www.scopus.com/pages/publications/85116069008. DOI: https://doi.org/10.1177/1748006X211048992.

40. Amrutkar, K. P. & Kamalja, K. K. "Efficient algorithm for reliability and importance measures of linear weighted-(n,f,k) and (n,f,k) systems". *Computers and Industrial Engineering*. 2017; 107: 85–99, https://www.scopus.com/pages/publications/85015035867. DOI: https://doi.org/10.1016/j.cie.2017.02.011.

41. Cui, L., Wang, M. & Jiang, W. "Reliability analysis of A combination of (n,f,k) and <n,f,k> systems". *Reliability Engineering and System Safety*. 2024; 249: 110191, https://www.scopus.com/pages/publications/85193451324. DOI: https://doi.org/10.1016/j.ress.2024.110191.

42. He, S., Wang, C., Wu, Z. & Guan, Q. "Reliability analysis for linear multi-state (n, f, k) and <n, f, k> systems". *Reliability Engineering and System Safety*. 2026; 272 (112245), https://www.scopus.com/pages/publications/105031574861. DOI: https://doi.org/10.1016/j.ress.2026.112245.

43. Triantafyllou, I. S. "Reliability Study of <n, f, 2> Systems: A Generating Function Approach". *International Journal of Mathematical, Engineering and Management Sciences*. 2021; 6 (1): 44–65, https://www.scopus.com/pages/publications/85097568436.
DOI: https://doi.org/10.33889/IJMEMS.2021.6.1.005.

44. Romankevitch, A., Morozov, K., Romankevich, V., Halytskyi, D. & Eleftherios, Z. "Improved GL-Model of Behavior of Complex Multiprocessor Systems in Failure Flow". *Lecture Notes on Data Engineering and Communications Technologies*. 2023; 181: 236-245, https://www.scopus.com/pages/publications/85169044963. DOI: https://doi.org/10.1007/978-3-031-36118-0_21.

Conflicts of Interest: The authors declare that they have no conflict of interest regarding this study, including financial, personal, authorship or other, which could influence the research and its results presented in this article

Received 06.04.2026
Received after revision 10.06.2926
Accepted 18.06.2026

DOI: https://doi.org/10.15276/aait.09.2026.25
УДК 004.05

Узагальнений метод побудови моделей поведінки відмовостійких багатопроцесорних систем у потоці відмов із міжпідсистемним резервуванням

Романкевич Віталій Олексійович[1)]
ORCID: https://orcid.org/0000-0003-4696-5935; zavkaf@scs.kpi.ua. Scopus Author ID: 57193263058
Морозов Костянтин В'ячеславович[1)]
ORCID: https://orcid.org/0000-0003-0978-6292; mcng@ukr.net. Scopus Author ID: 57222509251
Романкевич Олексій Михайлович[1)]
ORCID: https://orcid.org/0000-0001-5634-8469; romankev@scs.kpi.ua. Scopus Author ID: 6602114176
Галицький Даніїл Володимирович[1)]
ORCID: https://orcid.org/0009-0004-4421-3443; zipper135401@gmail.com. Scopus Author ID: 58553487600
Захаріудакіс Лефтеріс[2)]
ORCID: https://orcid.org/0000-0002-9658-3073; l.zacharioudakis@nup.ac.cy. Scopus Author ID: 57422876200
[1)] Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»,
пр. Перемоги, 37. Київ, 03056, Україна
[2)] Університет Неаполіс Пафос, Данаіс Авеню, 2. Пафос, 8042, Кіпр

АНОТАЦІЯ

Актуальність. Оцінка показників надійності відмовостійких багатопроцесорних систем є важливою задачею, зокрема під час їх аналізу та проєктування. Для її розв'язання можуть використовуватися GL-моделі поведінки систем у потоці відмов. При цьому для систем із міжпідсистемним резервуванням такі моделі повинні враховувати можливість використання процесорів підсистеми-донора як у роботоздатному стані цієї підсистеми, так і після втрати нею роботоздатності. Існуючі підходи не забезпечують повного врахування таких режимів функціонування. **Мета статті.** Розроблення узагальненого методу побудови GL-моделей поведінки в потоці відмов відмовостійких багатопроцесорних систем із міжпідсистемним резервуванням. **Завдання.** Формалізувати принципи побудови GL-моделей підсистеми-донора з урахуванням різних режимів використання її процесорів, розробити алгоритм реалізації запропонованого методу та провести експериментальну перевірку коректності побудованих моделей. **Методи.** Використано апарат GL-моделей, методи побудови базових і допоміжних GL-моделей, формування розширених векторів стану та аналізу зв'язності графів моделей. **Наукова новизна.** Запропоновано узагальнений метод побудови GL-моделей для відмовостійких багатопроцесорних систем із міжпідсистемним використанням резервних процесорів, який, на відміну від відомих підходів, враховує можливість використання процесорів підсистеми-донора як у разі збереження її роботоздатності, так і після втрати нею роботоздатності, коли роботоздатні процесори підсистеми можуть використовуватися для підтримання інших підсистем. **Практична значимість.** Запропонований метод може використовуватися для побудови GL-моделей складних відмовостійких багатопроцесорних систем та подальшої оцінки їх показників надійності. **Результати.** Розроблено метод і алгоритм побудови GL-моделей систем із міжпідсистемним резервуванням. Запропонований підхід ґрунтується на побудові допоміжних моделей, що описують набори резервних і роботоздатних процесорів підсистеми-донора, та формуванні розширених векторів стану для побудови моделей підсистем-реципієнтів. Розглянуто характерні приклади для систем з обмеженням кількості резервних ресурсів і систем із ковзним резервуванням. Проведені експериментальні дослідження підтвердили коректність побудованих моделей. Показано, що складність виразів реберних функцій отриманих GL-моделей є зіставною зі складністю відповідних базових моделей з еквівалентними параметрами відмовостійкості. **Висновки.** Запропонований метод забезпечує формальний опис міжпідсистемного резервування в GL-моделях та може бути інтегрований з іншими підходами до побудови моделей поведінки відмовостійких багатопроцесорних систем у потоці відмов. Результати експериментальних досліджень підтверджують коректність методу та доцільність його застосування в процесі аналізу й проєктування складних відмовостійких багатопроцесорних систем.

Ключові слова: відмовостійкі багатопроцесорні системи; GL-моделі; міжпідсистемне резервування; аналіз надійності; поведінка в потоці відмов; резервні ресурси

ABOUT THE AUTHORS

Vitaliy A. Romankevich - Doctor of Engineering Sciences, Professor, Head of System Programming and Specialized Computer System Department. National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Peremogy Ave. Kyiv, 03056, Ukraine

ORCID: https://orcid.org/0000-0003-4696-5935; zavkaf@scs.kpi.ua. Scopus Author ID: 57193263058

Research field: Fault-tolerant multiprocessor systems reliability estimation; GL-models; Self-diagnosable systems; Diagnosis of multiprocessor systems; Discrete mathematics

Романкевич Віталій Олексійович - доктор технічних наук, професор, завідувач кафедри Системного програмування і спеціалізованих комп'ютерних систем. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна

Kostiantyn V. Morozov - PhD, Assistant, System Programming and Specialized Computer System Department. National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, 37, Peremogy Ave. Kyiv, 03056, Ukraine
ORCID: https://orcid.org/0000-0003-0978-6292, mcng@ukr.net, Scopus Author ID: 57222509251

***Research field*:** GL-models; Fault-tolerant multiprocessor systems reliability estimation; Self-diagnosable multiprocessor systems

Морозов Костянтин В’ячеславович - кандидат технічних наук, доцент кафедри Системного програмування і спеціалізованих комп’ютерних систем. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна

Alexei M. Romankevich – Doctor of Engineering Sciences, Professor, System Programming and Specialized Computer System Department. National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, 37, Peremogy Ave. Kyiv, 03056, Ukraine
ORCID: https://orcid.org/0000-0001-5634-8469; romankev@scs.kpi.ua. Scopus Author ID: 6602114176

***Research field*:** Fault-tolerant multiprocessor systems reliability estimation; GL-models; Self-diagnosable systems; Diagnosis of multiprocessor systems; Multi-valued logic

Романкевич Олексій Михайлович - доктор технічних наук, професор, кафедра Системного програмування і спеціалізованих комп’ютерних систем. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна

Daniil V. Halytsky - postgraduate student, System Programming and Specialized Computer System Department, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, 37, Peremogy Ave. Kyiv, 03056, Ukraine
ORCID: https://orcid.org/0009-0004-4421-3443; zipper135401@gmail.com; Scopus Author ID: 58553487600

***Research field*:** GL-models building and processing optimization, Parallel computing and multi-core system optimization, Algorithm design for distributed and high-performance computing, Graph algorithms and their applications in complex systems, Computational efficiency in large-scale data processing.

Галицький Даніїл Володимирович - аспірант, кафедра Системного програмування і спеціалізованих комп’ютерних систем, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна

Lefteris Zacharioudakis - PhD (Eng), Assistant professor, Neapolis University Pafos, 2, Danais Ave. Pafos, 8042, Cyprus
ORCID: https://orcid.org/0000-0002-9658-3073; l.zacharioudakis@nup.ac.cy. Scopus Author ID: 57422876200

***Research field*:** Principles of CyberSecurity, Operating systems, Information Security, Cryptography, Penetration Testing

Захаріудакіс Лефтеріс - кандидат технічних наук, доцент. Університет Неаполіс Пафос, Данаіс Авеню, 2. Пафос, 8042, Кіпр