DOI: https://doi.org/10.15276/aait.07.2024.13
UDC 004.05

On the method of building of non-basic GL-models which are formed on combination of edge functions of basic models

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
Anna V. Morozova[1)]
ORCID: https://orcid.org/0009-0004-6738-8799; a.vl.morozova@gmail.com
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

This work is dedicated to the problem of building GL-models of behavior of non-basic fault-tolerant multiprocessor systems in the failure flow. Such models can be used to calculate the reliability parameters of the latter. The system, depending on the fulfillment of certain conditions, is resistant to failures of various multiplicities. These conditions depend only on the states of the system's processors and can be represented by special Boolean expressions. A method of constructing GL-models of such systems based on combining expressions of the edge functions of auxiliary basic models is proposed. At the same time, the specific feature of the models that were built by the proposed method is that they are based on cyclic graphs. This simplifies the process of evaluating their connectivity and also simplifies the analysis of the model's operation. In addition, this allows using other methods of modifying GL-models if necessary. The method involves using auxiliary models that have the same number of edges. In order to equalize the number of edges, auxiliary models can be extended with additional edges with edge functions of a special type. It is shown that this extension does not change the behavior of the models. In particular, the procedure of orthogonalization of the Boolean expressions is described, which should be carried out if the conditions can be satisfied simultaneously. It is shown that the expressions of the edge functions of the obtained GL-models, which can be quite complex, can sometimes be significantly simplified. Numerous experiments have been conducted to confirm the adequacy of the models (built by the proposed method) to the behavior of the corresponding systems in the failure flow. The example is given to demonstrate the application of the proposed method. The resulting model is analyzed and shown to correspond to the behavior of the system for which it was built.

Keywords: Fault-tolerant multiprocessor systems; reliability estimation; GL-models; k-out-of-n systems

For citation: **Romankevich V. A., Morozov K. V., Romankevich A. M., Morozova A. V., Zacharioudakis, L.** "**On the method of building of non-basic GL-models which are formed on combination of edge functions of basic models".** ***Applied Aspects of Information Technology.*** **2024; Vol. 7 No. 2: 175–188. DOI: https://doi.org/10.15276/aait.07.2024.13**

INTRODUCTION

In the modern world, more and more objects and systems are becoming automatic and automated. On the one hand, this allows us to free people from performing monotonous and non-creative tasks, and on the other hand, to improve the quality of process management by reducing the influence of the human factor. One of the key components of such objects is their control system (CS) [1, 2].

In certain cases, especially for so-called critical application systems (CAS) [3, 4], [5], the breakdown of which can lead to significant consequences, additional reliability requirements may be imposed on the control systems (CS). Furthermore, the management of such objects often involves solving problems with a rather high computational complexity. The use of fault-tolerant multiprocessor systems (FTMS) [6, 7], [8, 9], [10, 11], [12] as CS allows achieving the required level of both reliability and performance [13, 14].

Sooner or later, developers of fault-tolerant multiprocessor systems (FTMS) face the task of estimating its reliability parameters, such as the probability of failure. For complex and heterogeneous systems, such as control systems (CS), solving this problem can be very challenging [15, 16], [17, 18].

There are two main groups of methods for solving it [16, 19], [20, 21].

The methods of the first group are based on building of analytical expressions for calculating the relevant system reliability parameters based on the corresponding parameters of its processor components [22, 23], [24, 25]. The advantage of such approaches is the relative simplicity of the calculation process itself, as well as the ability to achieve high calculation accuracy. The disadvantage, is the complexity of building these expressions for different types of systems: for each of them, a new calculation method often has to be created [23, 26], [27, 28], [29, 30], [31, 32], [33, 34], [35, 36], [37, 38], [39, 40], [41, 42], [43, 44], [45, 46], [47, 48], [49, 50], [51, 52], [53, 54].

The methods of the second group are based on statistical experiments with models of the behavior of the fault-tolerant multiprocessor systems (FTMS) in the failure flow [55, 56]. Their advantage is their flexibility, while the disadvantage is the need to conduct a large number of experiments with models. At the same time, the accuracy of estimating the system reliability parameters in general depends on the number of experiments conducted. Therefore, reducing the complexity of each experiment allows to achieve greater accuracy of calculations in practice.

GL-models [56, 57] can be used as models of behavior of fault-tolerant multiprocessor systems (FTMS) in the failure flow. A GL-model is an undirected graph, each edge of which corresponds to a Boolean edge function. The arguments of the edge functions are elements of the so-called system state vector, a Boolean vector that reflects the state of each of its components (processors) in the failure flow. If the processor is functional, the corresponding element of the state vector takes the value equal to 1, and if it is not functional, it takes the value equal to 0. If the edge function takes a zero value, the corresponding edge is excluded from the graph. The connectivity of the model graph for a certain vector reflects the overall state of the system. For a given combination of functional and non-functional processors, a connected graph indicates that the system remains operational, while the loss of connectivity in the graph corresponds to a system breakdown.

There are a few methods for building GL-models [58, 59], [60, 61], which typically can be used to build models of so-called basic systems. Systems that are resistant to any failure with a multiplicity not exceeding a certain threshold but are not resistant to any failure of greater multiplicity are called basic systems. Models of such systems are also called basic models. A basic system, consisting of n processors and resistant to failure of no more than m of them, as well as its corresponding model, is denoted by $K(m, n)$.

Some of the methods can be used to build basic GL-models that are based on cyclic graphs [58, 59]. The advantage of such models is that the process of determining the graph's connectivity is greatly simplified.

It is worth noting that real systems, especially control systems, are not always basic, or in other words, they are non-basic. Such systems may be resistant only to a certain set of failures of a particular multiplicity, but not to other failures of the same multiplicity.

The building of non-basic GL-models (i.e., models of non-basic systems) can be performed by modifying basic ones. This modification can occur by changing the structure of the graph [62], by modifying the expressions of the edge functions [63], or by combining both approaches.

FORMULATION OF THE PROBLEM

The application of modification methods of GL-models that do not change the structure of the graph [63] to GL-models based on cyclic graphs [58, 59] allows simplifying the procedure of analyzing the model's operation in general. This, in turn, leads to a reduction in the time required to perform one experiment with the model [64]. However, this problem has not yet been solved in general case (for an arbitrary non-basic system).

Therefore, the problem of building a GL-model of an arbitrary non-basic FTMS while preserving the cyclic structure of the graph is relevant.

MLE-MODELS

Let us take a closer look at one of the types of GL-models that lose a minimum number of edges on vectors with increased multiplicity of zeros – the so-called MLE-models (minimum of lost edges) [59]. Such models can be built for basic $K(m, n)$ systems with arbitrary values of m and n.

They are based on cyclic graphs and have a few special properties [59, 65]. For example, the MLE-model $K(m, n)$ contains exactly $\varphi(m, n) = n - m + 1$ edges and always loses exactly $\psi(m, l)$ of them on vectors with l zeros, where

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

The building of the expressions of the edge functions of such a model is done by dividing the input vector into two parts (usually of equal or nearly equal size, but in general, such a division can be arbitrary).

Let us denote their sizes as n_1 and n_2, $n_1 + n_2 = n$.

Then the edge functions of the model will consist of [59]:

1) The edge functions of the model $K_1(m, n_1)$, constructed for the first part of the vector.

2) The edge functions of the form $f = \kappa_1(i, n_1) \vee \kappa_2(m - i, n_2)$, where $i = 1, 2, \ldots, m - 1$, and the expressions $\kappa_1(i, n_1)$ and $\kappa_2(m - i, n_2)$ are the conjunctions of the expressions of all edge functions of the models $K_1(i, n_1)$ and $K_2(m - i, n_2)$, respectively, built for the first and second part of the input vector.

3) The edge functions of the model $K_2(m, n_2)$, built for the second part of the vector.

The abovementioned auxiliary models are built in a similar way until models of the form $K(1, n)$ or $K(n, n)$, are reached, the expressions of the edge functions of which are known. The $K(1, n)$ model contains exactly n functions of the form $f_i(X) = x_i$, where $i = 1, 2, \ldots, n$, and x_i – are elements of the input system state vector X, i.e., $X = \langle x_1, x_2, \ldots, x_n \rangle$ The $K(n, n)$ model, in turn, contains only one edge function of the form $f(X) = x_1 \vee x_2 \vee \ldots \vee x_n$. If some of the auxiliary models of the form $K(a, b)$ cannot be obtained in principle (which happens if $a \geq b$), then the corresponding edge functions, for the synthesis of which such models are necessary according to the above algorithm, are not included in the set of edge functions of the model being built.

RESTRICTIONS

A non-basic fault-tolerant multiprocessor system differs from a basic one in that it is resistant to only certain failures of a specific multiplicity, but fails in the event of other failures of the same multiplicity. In general, many different configurations of such systems are possible. However, from a practical point of view, there are several obvious restrictions. For example, if an FTMS is resistant to a certain failure, it can be expected to be resistant to all lower multiplicity failures, that can be obtained from that failure by replacing some failed processors with functional ones.

Practically speaking, it is also often the case that a non-basic FTMS differs from a basic one precisely, because it remains stable over a certain set of failures of increased multiplicity. Therefore, it can be said that the system generally behaves like a basic $K(m, n)$, but under certain conditions, like a basic $K(m + t, n)$. In a more general case, there may be more than one such condition and value of t. In other words, the system can be considered such that under some conditions C_i behaves like the basic $K(m_i, n)$, where $i = 1, 2, \ldots, k$. At the same time, it is expected that abovementioned conditions depend only on the states of specific processors of the system, and therefore can be expressed in form of some Boolean expressions $c_i(X)$, where X is the system state vector. This expression takes the value of 1 for those system state vectors in which the corresponding condition C_i is satisfied and 0 otherwise. This is the case that will be discussed in this article.

EXTENSION OF MLE-MODELS

As mentioned above, MLE-models can be built for any $K(m, n)$ system. One of the advantages of such models is that they are all based on cyclic graphs. The number of edges in such a graph is $\varphi(m, n) = n - m + 1$.

The system under consideration, depending on the fulfillment of conditions, can behave as a basic model with different values of the degree of fault tolerance (value of m). For each of these cases, a different MLE-model can be built. In this case, their graphs will obviously differ in the number of edges.

To further build a model of the system as a whole, it will be necessary to obtain models with the same number of edges. Fortunately, this can be done in many different methods.

One of the simplest methods is to add extra edges to the model with trivial edge functions of the form $f \equiv 1$. Since the remaining edge functions will remain the same as in the original model, the resulting model will lose exactly $\psi(m, l)$ edges on vectors with l zeros, just like the original model. Moreover, since the graph of the model will remain cyclic, it will lose its connectivity exactly at the same time as the graph of the original model, that is, in the case of the loss of any two or more edges.

Therefore, all models can be reduced to the same number of edges by simply adding edges with functions of the form $f \equiv 1$.

THE COMBINED MODEL

Suppose there is a set of conditions $C_1, C_2, \ldots, C_k$, each of which corresponds to the behavior in the failure flow of the non-basic FTMS to basic $K_1(m_1, n), K_2(m_2, n), \ldots, K_k(m_k, n)$ systems, respectively. At the same time, exactly one condition is always satisfied.

Let us assume that for each of the basic systems $K_1(m_1, n), K_2(m_2, n), \ldots, K_k(m_k, n)$, GL-models are built in some way, where each of them is based on a cyclic graph. In addition, the graphs of all models contain the same number of edges, denoted as L. The edge functions of these models are denoted as f_j^i, where $i = 1, 2, \ldots, k$, and $j = 1, 2, \ldots, L$.

We will build a GL-model based on a cyclic graph with L edges, with the following edge functions:

$$f_j(X) = \bigvee_{i=1}^{k} c_i(X) f_j^i(X),$$

where $j = 1, 2, \ldots, L$. We will demonstrate that such a model indeed corresponds to the behavior in the failure stream of the previously described system.

Let us consider the situation (corresponding to vector X), when some condition C_h is satisfied, that is, $c_h(X) = 1$. Given that exactly one condition is satisfied at a time, we get $c_i(X) = 0$ for all $i \neq h$.

Thus, the edge functions of the model will take the form

$$f_j(X) = f_j^h(X).$$

It is easily seen that in this case, the model will completely coincide with the h-th model, which corresponds to the system $K_h(m_h, n)$. Therefore, for each of the conditions C_i the model will indeed correspond to the system $K_i(m_i, n)$, as required.

ORTHOGONALIZATION OF THE CONDITIONS

In the previous section, it was assumed that exactly one of the conditions $C_1, C_2, \ldots, C_k$ must always be satisfied. However, in practice, the conditions can be formulated in way that some of them will be satisfied simultaneously on certain vectors. For example, when describing a system, it can be stated that it is m-fault-tolerant, and when a certain condition is met, it becomes $(m + 1)$-fault-tolerant. In other words, considering that an $(m + t)$-fault-tolerant system is also resistant to all failures of multiplicity no higher than m, those to which an m-fault-tolerant system is resistant, it is reasonable to assume that in the case of simultaneous satisfaction of multiple conditions, the system behaves as one with the highest degree of fault tolerance (of those for which the conditions are met).

The application of the proposed approach in case when multiple conditions are met simultaneously on certain system state vectors can lead to undesirable side effects, that lead to a loss of adequacy of the model to the system behavior. For example, let the conditions $C_{T_1}, C_{T_2}, \ldots, C_{T_L}$ be satisfied on a certain vector.

Then, the edge functions of the model obtained by the proposed method will be as follows:

$$f_j = f_j^{T_1} \vee f_j^{T_2} \vee \ldots \vee f_j^{T_L}.$$

Therefore, in general, the model will differ from the basic model by the expressions of its edge functions. The possibility of side effects will depend on the method and parameters of building auxiliary models, as well as the numbering order of their edge functions. For instance, it is possible that vectors with an increased multiplicity of zeros will be blocked (i.e., the model will indicate the system's working state on them).

In order to be able to use the proposed method of building GL-models for non-basic FTMS, even if the conditions are not mutually exclusive, it is enough to first perform a specialized procedure for orthogonalizing the expressions of these conditions. After that, exactly one condition will be satisfied for each input vector, and the proposed method can be applied. Let us consider this procedure in more detail.

Suppose that a non-basic FTMS, under conditions $C_1, C_2, \ldots, C_k$ is fault-tolerant of multiplicity no higher than $m_1, m_2, \ldots, m_k$ respectively. In this case, at least one of the above conditions is always satisfied, but some of them can be satisfied simultaneously. Also, we will assume that $m_1 < m_2 < \ldots < m_k$ (we can always renumber them so that is satisfied).

Let us form conditions $S_1, S_2, \ldots, S_k$ so that the condition S_i is satisfied if and only if the condition C_i is also satisfied, but none of the conditions with a higher index are satisfied.

Boolean expressions that correspond to such conditions can be built as

$$s_i(X) \equiv c_i(X) \wedge \bigwedge_{j=i+1}^{k} \bar{c}_j(X),$$

for $i = 1, 2, \ldots, k - 1$, and $s_k(X) \equiv c_k(X)$.

In other words, if condition S_i is satisfied for a certain vector, then i is the largest index among the conditions $C_1, C_2, \ldots, C_k$, that are satisfied for this vector. It is easy to see that with this construction, exactly one of the conditions $S_1, S_2, \ldots, S_k$ will always be satisfied (considering that at least one of the conditions $C_1, C_2, \ldots, C_k$ is always satisfied).

The proposed replacement is correct, since, as mentioned earlier, the model $K(m_a, n)$ is resistant to all failures, including all failures to which the model $K(m_b, n)$, is also resistant if $m_a < m_b$.

ALGORITHM FOR BUILDING A MODEL OF A NON-BASIC SYSTEM

Now we can formulate the algorithm for building a GL-model of a non-basic system. Let us recall that we consider a non-basic FTMS, which, under the conditions $C_1, C_2, \ldots, C_k$ is resistant to failures of multiplicity not higher than $m_1, m_2, \ldots, m_k$ respectively.

Moreover, the conditions $C_1, C_2, \dots, C_k$ depend only on the states in the failure flow of the processors of the system (all or some of them). In addition, at least one of these conditions always satisfied. The building of the GL-model of such a system will consists of the following steps.

Step 1. Building logical expressions of conditions. For each of the conditions C_i we build the corresponding Boolean expressions $c_i(X)$, which depend on the values of the elements of the state vector of the system X.

Step 2. Rearrangement. If for the values $m_1, m_2, \dots, m_k$, the condition $m_1 < m_2 < \dots < m_k$ is not satisfied, then we rearrange them so as to satisfy it. In other words, we find a set of indices $T = \langle t_1, t_2, \dots, t_k \rangle \in \mathfrak{S}_k$, for which $m_{t_1} < m_{t_2} < \dots < m_{t_k}$, where $\mathfrak{S}_k$ – is the set of all possible permutations of numbers 1, 2, …, k.

Step 3. Orthogonalization of logical expressions of conditions. If some conditions can be satisfied simultaneously, we build orthogonalized logical expressions $s_{t_i}(X) \equiv c_{t_i}(X) \wedge \bigwedge_{j=i+1}^{k} \bar{c}_{t_j}(X)$, for $i = 1, 2, \dots, k-1$, and $s_{t_k}(X) \equiv c_{t_k}(X)$.

Step 4. Building of MLE-models. We build basic MLE-models $K_i(m_i, n)$ for $i = 1, 2, \dots, k$. Their functions will be denoted as $f_{j_i}^{i}$, where $j_i = 1, 2, \dots, \varphi(m_i, n)$, where $\varphi(m_i, n) = n - m_i + 1$.

Step 5. Extension of MLE-models. We supplement all obtained MLE-models to the same number of edge functions by adding trivial functions of the form $f \equiv 1$. Considering that the value of m_{t_1} is the smallest, the model $K_{t_1}(m_{t_1}, n)$ will have the largest number of edges, namely $L = \varphi(m_{t_1}, n) = n - m_{t_1} + 1$. Therefore, it is sufficient to supplement all models except the t_1-th, model with additional $m_i - m_{t_1}$ edge functions $f_{j_i}^{i} \equiv 1$, where $j_i = \varphi(m_i, n) + 1, \varphi(m_i, n) + 2, \dots, L$. As a result, all models will contain exactly L edge functions, denoted as f_j^i, where $i = 1, 2, \dots, k$ is the model (and condition) number, and $j = 1, 2, \dots, L$ is function number.

Step 6. Building expressions of edge function of the model. We build the expressions of the edge functions of the model according to the formula

$$f_j(X) = \bigvee_{i=1}^{k} s_i(X) f_j^i(X),$$

where $j = 1, 2, \dots, L$ is function number.

Therefore, the result is a GL-model that is based on a cyclic graph with L edges and edge functions with expressions $f_j(X)$, where $j = 1, 2, \dots, L$.

Sometimes, they obtained expressions of edge functions can be further simplified. Moreover, the edge functions in each of the models can be rearranged in any order, allowing them to be arranged, so that such simplification is more effective.

EXAMPLE

As an example, let us build a GL-model for a system that contains 9 processors and is resistant to the failure of any 3 of them if the 1st and 2nd, or 5th processors are operational, and also resistant to the failure of any 4 of them if the 4th, 5th, and 7th processors are simultaneously functional, and in all other cases the system is 2-fault tolerant. To do this, follow the steps described in the previous section. As you can see, based on the problem, we have, $k = 3$, $m_1 = 3$, $m_2 = 4$, $m_3 = 2$.

Step 1. Let us build logical expressions for the conditions. Condition C_1 is satisfied when 1st and 2nd, or 5th processors are operational, and it corresponds to the expression $c_1(X) = x_1 x_2 \vee x_5$. Condition C_2 is satisfied when 4th, 5th, and 7th processors are operational, and it corresponds to the expression $c_2(X) = x_4 x_5 x_7$. Condition C_3 is always satisfied, so its expression is simply $c_3(X) = 1$.

Step 2. Since the condition $m_1 < m_2 < m_3$ is not satisfied, we need to rearrange these values. The correct relation is $m_3 < m_1 < m_2$, so the desired set of indices is $T = \langle 3, 1, 2 \rangle$.

Step 3. Conditions C_1, C_2 and C_3 can occur simultaneously, so we can perform orthogonalization. Accordingly, we have:

$$s_3(X) = c_3(X)\bar{c}_1(X)\bar{c}_2(X) = 1 \wedge \left(\overline{x_1 x_2 \vee x_5}\right) \wedge$$
$$\wedge \left(\overline{x_4 x_5 x_7}\right) = \left(\left(\overline{x_1 x_2}\right)\bar{x}_5\right)(\bar{x}_4 \vee \bar{x}_5 \vee \bar{x}_7) =$$
$$= (\bar{x}_1 \vee \bar{x}_2)\bar{x}_5(\bar{x}_4 \vee \bar{x}_5 \vee \bar{x}_7) =$$
$$= \bar{x}_5(\bar{x}_1 \vee \bar{x}_2);$$
$$s_1(X) = c_1(X)\bar{c}_2(X) = (x_1 x_2 \vee x_5)\left(\overline{x_4 x_5 x_7}\right) =$$
$$= (x_1 x_2 \vee x_5)(\bar{x}_4 \vee \bar{x}_5 \vee \bar{x}_7) =$$
$$= x_1 x_2 \bar{x}_5 \vee x_5(\bar{x}_4 \vee \bar{x}_7);$$
$$s_2(X) = c_2(X) = x_4 x_5 x_7.$$

Step 4. Let us build the MLE-models $K_1(3,9)$, $K_2(4, 9)$ and $K_3(2, 9)$.

The $K_1(3, 9)$ model has the following 7 edge functions (Fig. 1a):

$$f_1^1 = x_1 \vee x_2 \vee x_3;$$
$$f_2^1 = (x_1 \vee x_2)(x_1 x_2 \vee x_3) \vee x_4 x_5;$$
$$f_3^1 = x_1 x_2 x_3 \vee x_4 \vee x_5;$$
$$f_4^1 = (x_1 \vee x_2)(x_1 x_2 \vee x_3)(x_1 x_2 x_3 \vee x_4 x_5) \wedge$$
$$\wedge (x_4 \vee x_5) \vee x_6 x_7 x_8 x_9;$$
$$f_5^1 = x_1 x_2 x_3 x_4 x_5 \vee$$
$$\vee (x_6 \vee x_7)(x_6 x_7 \vee x_8 x_9)(x_8 \vee x_9);$$
$$f_6^1 = x_6 \vee x_7 \vee x_8 x_9;$$

$f_7^1 = x_6x_7 \lor x_8 \lor x_9$.

The $K_2(4, 9)$ model contains 6 edge functions (Fig. 1b):

$f_1^2 = x_1 \lor x_2 \lor x_3 \lor x_4x_5$;

$f_2^2 = (x_1 \lor x_2)(x_1x_2 \lor x_3) \lor x_4 \lor x_5$;

$f_3^2 = (x_1 \lor x_2 \lor x_3) \land$
$\land \big((x_1 \lor x_2)(x_1x_2 \lor x_3) \lor x_4x_5\big) \land$
$\land (x_1x_2x_3 \lor x_4 \lor x_5) \lor x_6x_7x_8x_9$;

$f_4^2 = (x_1 \lor x_2)(x_1x_2 \lor x_3)(x_1x_2x_3 \lor x_4x_5) \land$
$\land (x_4 \lor x_5) \lor (x_6 \lor x_7)(x_6x_7 \lor x_8x_9)(x_8 \lor x_9)$;

$f_5^2 = x_1x_2x_3x_4x_5 \lor$
$\lor (x_6 \lor x_7 \lor x_8x_9)(x_6x_7 \lor x_8 \lor x_9)$;

$f_6^2 = x_6 \lor x_7 \lor x_8 \lor x_9$.

In turn, the $K_3(2, 9)$ model includes exactly 8 edge functions (Fig. 1c):

$f_1^3 = x_1 \lor x_2$;

$f_2^3 = x_1x_2 \lor x_3$;

$f_3^3 = x_1x_2x_3 \lor x_4x_5$;

$f_4^3 = x_4 \lor x_5$;

$f_5^3 = x_1x_2x_3x_4x_5 \lor x_6x_7x_8x_9$;

$f_6^3 = x_6 \lor x_7$;

$f_7^3 = x_6x_7 \lor x_8x_9$;

$f_8^3 = x_8 \lor x_9$.

Step 5. Let us supplement the models to have the same number of edges. To do this, we will add one additional edge function $f_8^1 = 1$ to model $K_1(3, 9)$, and two additional edge functions $f_7^2 = 1$ and $f_8^2 = 1$ to model $K_2(4, 9)$. Now all the models contain exactly 8 edges (Fig. 2).

Step 6. Using the expressions of the edge functions of the models f_j^i and the expressions of the conditions s_i we will build the edge functions of the GL-model of the system under consideration. This model will be based on a cyclic graph with 8 edges and the following edge functions:

$f_1 = s_1f_1^1 \lor s_2f_1^2 \lor s_3f_1^3 =$
$= \big(x_1x_2\bar{x}_5 \lor x_5(\bar{x}_4 \lor \bar{x}_7)\big)(x_1 \lor x_2 \lor x_3) \lor$
$\lor x_4x_5x_7(x_1 \lor x_2 \lor x_3 \lor x_4x_5) \lor$
$\lor \bar{x}_5(\bar{x}_1 \lor \bar{x}_2)(x_1 \lor x_2)$;

$f_2 = s_1f_2^1 \lor s_2f_2^2 \lor s_3f_2^3 =$
$= \big(x_1x_2\bar{x}_5 \lor x_5(\bar{x}_4 \lor \bar{x}_7)\big)((x_1 \lor x_2)(x_1x_2 \lor x_3)$
$\lor$
$\lor x_4x_5) \lor x_4x_5x_7 \land$
$\land \big((x_1 \lor x_2)(x_1x_2 \lor x_3) \lor x_4 \lor x_5\big) \lor$
$\lor \bar{x}_5(\bar{x}_1 \lor \bar{x}_2)(x_1x_2 \lor x_3)$;

$f_3 = s_1f_3^1 \lor s_2f_3^2 \lor s_3f_3^3 =$
$= \big(x_1x_2\bar{x}_5 \lor x_5(\bar{x}_4 \lor \bar{x}_7)\big)(x_1x_2x_3 \lor x_4 \lor x_5) \lor$
$\lor x_4x_5x_7((x_1 \lor x_2 \lor x_3)((x_1 \lor x_2)(x_1x_2 \lor x_3) \lor$
$\lor x_4x_5)(x_1x_2x_3 \lor x_4 \lor x_5) \lor x_6x_7x_8x_9) \lor$
$\lor \bar{x}_5(\bar{x}_1 \lor \bar{x}_2)(x_1x_2x_3 \lor x_4x_5)$;

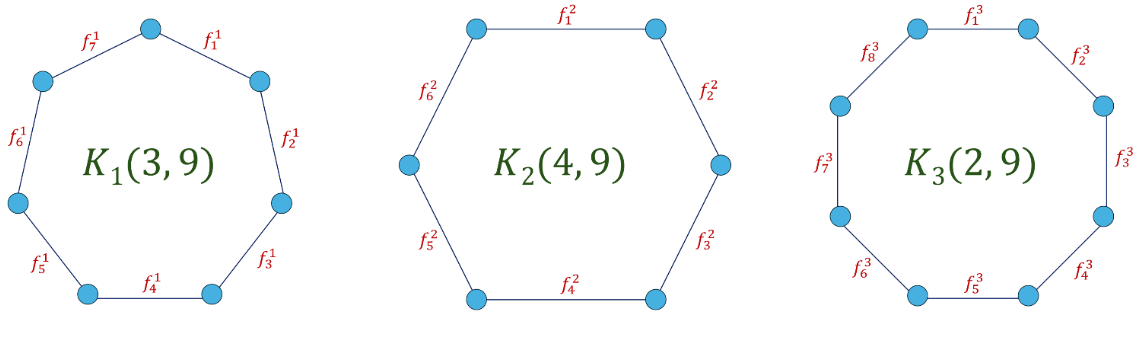

Fig. 1. **Auxiliary MLE-models: a – $K_1(3, 9)$; b – $K_2(4, 9)$; c – $K_3(2, 9)$**
***Source:* compiled by the authors**

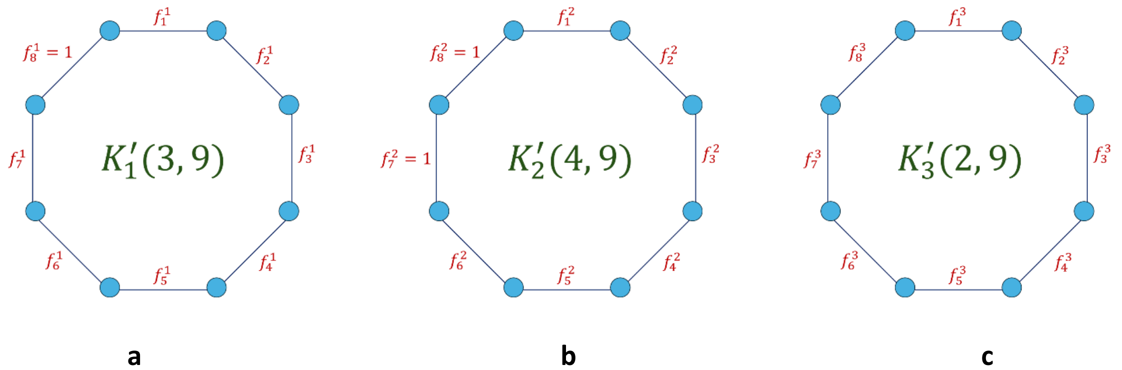

Fig. 2. **Auxiliary MLE-models after extension step: a – $K_1(3, 9)$; b – $K_2(4, 9)$; c – $K_3(2, 9)$**

Source: **compiled by the authors**

$f_4 = s_1 f_4^1 \vee s_2 f_4^2 \vee s_3 f_4^3 =$
$= \left(x_1 x_2 \bar{x}_5 \vee x_5(\bar{x}_4 \vee \bar{x}_7)\right)((x_1 \vee x_2) \wedge$
$\wedge (x_1 x_2 \vee x_3)(x_1 x_2 x_3 \vee x_4 x_5)(x_4 \vee x_5) \vee$
$\vee x_6 x_7 x_8 x_9) \vee x_4 x_5 x_7((x_1 \vee x_2)(x_1 x_2 \vee x_3) \wedge$
$\wedge (x_1 x_2 x_3 \vee x_4 x_5)(x_4 \vee x_5) \vee (x_6 \vee x_7) \wedge$
$\wedge (x_6 x_7 \vee x_8 x_9)(x_8 \vee x_9)) \vee \bar{x}_5(\bar{x}_1 \vee \bar{x}_2) \wedge$
$\wedge (x_4 \vee x_5);$

$f_5 = s_1 f_5^1 \vee s_2 f_5^2 \vee s_3 f_5^3 =$
$= \left(x_1 x_2 \bar{x}_5 \vee x_5(\bar{x}_4 \vee \bar{x}_7)\right)(x_1 x_2 x_3 x_4 x_5 \vee$
$\vee (x_6 \vee x_7)(x_6 x_7 \vee x_8 x_9)(x_8 \vee x_9)) \vee$
$\vee x_4 x_5 x_7(x_1 x_2 x_3 x_4 x_5 \vee (x_6 \vee x_7 \vee x_8 x_9) \wedge$
$\wedge (x_6 x_7 \vee x_8 \vee x_9)) \vee \bar{x}_5(\bar{x}_1 \vee \bar{x}_2) \wedge$
$\wedge (x_1 x_2 x_3 x_4 x_5 \vee x_6 x_7 x_8 x_9);$

$f_6 = s_1 f_6^1 \vee s_2 f_6^2 \vee s_3 f_6^3 =$
$= \left(x_1 x_2 \bar{x}_5 \vee x_5(\bar{x}_4 \vee \bar{x}_7)\right)(x_6 \vee x_7 \vee x_8 x_9) \vee$
$\vee x_4 x_5 x_7(x_6 \vee x_7 \vee x_8 \vee x_9) \vee$
$\vee \bar{x}_5(\bar{x}_1 \vee \bar{x}_2)(x_6 \vee x_7);$

$f_7 = s_1 f_7^1 \vee s_2 f_7^2 \vee s_3 f_7^3 =$
$= \left(x_1 x_2 \bar{x}_5 \vee x_5(\bar{x}_4 \vee \bar{x}_7)\right)(x_6 x_7 \vee x_8 \vee x_9) \vee$
$\vee x_4 x_5 x_7 \vee \bar{x}_5(\bar{x}_1 \vee \bar{x}_2)(x_6 x_7 \vee x_8 x_9);$

$f_8 = s_1 f_8^1 \vee s_2 f_8^2 \vee s_3 f_8^3 =$
$= \left(x_1 x_2 \bar{x}_5 \vee x_5(\bar{x}_4 \vee \bar{x}_7)\right) \vee x_4 x_5 x_7 \vee$
$\vee \bar{x}_5(\bar{x}_1 \vee \bar{x}_2)(x_8 \vee x_9).$

The model is presented in the Fig. 3.

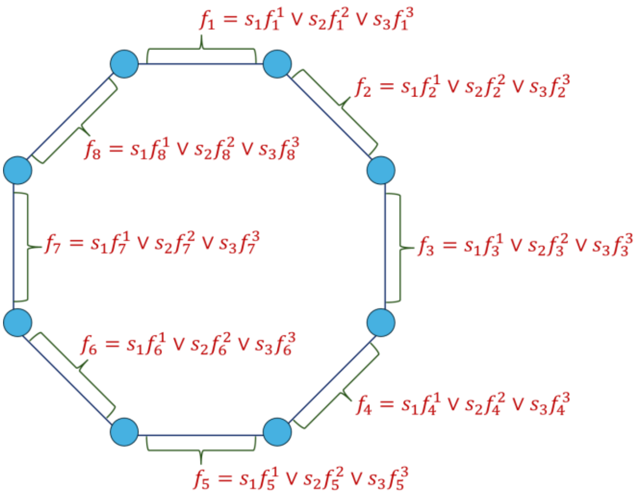

Fig. 3. **Non-basic GL-model built with the proposed method**
Source: **compiled by the authors**

Note that the obtained expressions of the edge function can be simplified. For example, the expression of the function f_1 after simplification will be as follows:

$$f_1 = x_1 \vee x_2 \vee x_5(x_3 \vee x_4 x_7).$$

According to the experimental results, it has been established that such a model shows an operational state of the system for all vectors with no more than 2 zeros, as well as for the next 71 vectors with 3 zeros: 111111000, 111110100, 111101100, 111011100, 110111100, 101111100, 011111100, 111110010, 111101010, 111011010, 110111010, 101111010, 011111010, 111100110, 111010110, 110110110, 101110110, 011110110, 111001110, 110101110, 110011110, 101011110, 011011110, 100111110, 010111110, 001111110, 111110001, 111101001, 111011001, 110111001, 101111001, 011111001, 111100101, 111010101, 110110101, 101110101, 011110101, 111001101, 110101101, 110011101, 101011101, 011011101, 100111101, 010111101, 001111101, 111100011, 111010011, 110110011, 101110011, 011110011, 111001011, 110101011, 110011011, 101011011, 011011011, 100111011, 010111011, 001111011, 111000111, 110100111, 110010111, 101010111, 011010111, 100110111, 010110111, 001110111, 110001111, 100011111, 010011111, 001011111, 000111111; and also on the following 15 vectors with 4 zeros: 110110100, 101110100, 011110100, 100111100, 010111100, 001111100, 100110110, 010110110, 001110110, 000111110, 100110101, 010110101, 001110101, 000111101, 000110111.

Note that the above set of vectors with 3 zeros include all possible vectors of length 9 with 3 zeros, in which the 1st and 2nd elements have a value equal to 1 (which corresponds to situations where the 1st and 2nd processors are operational), namely the following $C_{9-2}^3 = 35$ vectors: **11**1111000, **11**1110100, **11**1101100, **11**1011100, **11**0111100, **11**1110010, **11**1101010, **11**1011010, **11**0111010, **11**1100110, **11**1010110, **11**0110110, **11**1001110, **11**0101110, **11**0011110, **11**1110001, **11**1101001, **11**1011001, **11**0111001, **11**1100101, **11**1010101, **11**0110101, **11**1001101, **11**0101101, **11**0011101, **11**1100011, **11**1010011, **11**0110011, **11**1001011, **11**0101011, **11**0011011, **11**1000111, **11**0100111, **11**0010111, **11**0001111; as well as all possible vectors of length 9 with 3 zeros, in which the 5th element has a value of 1 (which corresponds to the functionality of the 5th processor), namely the following $C_{9-1}^3 = 56$ vectors: 1111**1**1000, 1111**1**0100, 1110**1**1100, 1101**1**1100, 1011**1**1100, 0111**1**1100, 1111**1**0010, 1110**1**1010, 1101**1**1010, 1011**1**1010, 0111**1**1010, 1110**1**0110, 1101**1**0110, 1011**1**0110, 0111**1**0110, 1100**1**1110, 1010**1**1110, 0110**1**1110, 1001**1**1110, 0101**1**1110, 0011**1**1110, 1111**1**0001, 1110**1**1001, 1101**1**1001, 1011**1**1001, 0111**1**1001, 1110**1**0101, 1101**1**0101, 1011**1**0101, 0111**1**0101, 1100**1**1101, 1010**1**1101, 0110**1**1101, 1001**1**1101, 0101**1**1101, 0011**1**1101, 1110**1**0011, 1101**1**0011, 1011**1**0011, 0111**1**0011, 110011011, 101011011, 011011011, 100111011, 010111011, 001111011, 110010111, 101010111, 011010111, 100110111, 010110111, 001110111, 100011111, 010011111, 001011111, 000111111. These vectors correspond exactly to situations when the 1st and 2nd or the 5th processors are operational at the same time, making the system 3-failure tolerant according to the condition. It can be seen that among these vectors, there are those that are contained in both sets. These are the vectors in which both the 1st, 2nd and the 5th elements have a value of 1 (i.e., both parts of the condition are simultaneously satisfied: both the 1st and 2nd processors, and the 5th processor are functional), namely the following $C_{9-2-1}^{3}=20$ vectors: **11**11**1**1000, **11**11**1**0100, **11**01**1**1100, **11**01**1**1100, **11**11**1**0010, **11**01**1**1010, **11**01**1**1010, **11**10**1**0110, **11**01**1**0110, **11**00**1**1110, **11**11**1**0001, **11**01**1**1001, **11**01**1**1001, **11**10**1**0101, **11**01**1**0101, **11**00**1**1101, **11**10**1**0011, **11**01**1**0011, **11**00**1**1011, **11**00**1**0111. Therefore, the total number of vectors with 3 zeros, on which the model demonstrates the system's operable state is indeed 35 + 56 – 20 = 71.

It should be noted that this set would also include those vectors for which the condition is met, that the system is resistant to failures of a higher-order. For this system, such a condition is the functionality of the 4th, 5th, and 7th processors simultaneously, under which the system is 4-fault tolerant. This condition is met by all possible vectors of length of 9 with 3 zeros, in which the 4th, 5th, and 7th elements have a value of 1, namely the following $C_{9-3}^{3}=20$ vectors: 111**11**0**1**00, 110**11**1**1**00, 101**11**1**1**00, 011**11**1**1**00, 110**11**0**1**10, 101**11**0**1**10, 011**11**0**1**10, 100**11**1**1**10, 010**11**1**1**10, 001**11**1**1**10, 110**11**0**1**01, 101**11**0**1**01, 011**11**0**1**01, 100**11**1**1**01, 010**11**1**1**01, 001**11**1**1**01, 100**11**0**1**11, 010**11**0**1**11, 001**11**0**1**11, 000**11**1**1**11.

However, obviously, all of these vectors are included in the previously mentioned set of vectors with the 5th element equal to 1.

As for the set of vectors with 4 zeros, there are indeed all the vectors of length 9 with 4 zeros, in which the 4th, 5th, and 7th elements have the value 1, which corresponds to the condition where the system is 4-fault-tolerant, namely, the simultaneous operability of the 4th, 5th, and 7th processors. The number of such vectors is indeed $C_{9-3}^{4}=15$.

So, as we can see, the built GL-model adequately reflects the behavior in the failure flow of the considered system for which it was built.

CONCLUSIONS

This article solves the problem of building GL-models of behavior in the failure flow of non-basic fault-tolerant multiprocessor systems. These include, in particular, fault-tolerant aircraft control systems, which sometimes involve hundreds of processors. Depending on certain conditions, such a system is assumed to be resistant to failures of various multiplicity. These conditions always depend only on the states of the system's processors.

The proposed method of building GL-models is reduced to combining the expressions of edge functions of special auxiliary basic models and logical expressions correspond to the aforementioned conditions. The resulting GL-models are based on cyclic graphs, which simplifies the analysis of model during experimentation. In addition, this allows for the additional use of other methods of model modification, if necessary.

Since the auxiliary models can have a different number of edges in a graph, they can be supplemented with additional edges to equalize the number of the latter. It has been shown that the addition of such edges with edge functions of a special form does not change the behavior of the model.

To apply the proposed method, exactly one of the conditions that define each of the fault-tolerance levels of the system must always be met. If this is not the case, then an additional procedure of orthogonalization of the condition expressions can be performed to make them so.

Numerous experiments have been conducted, to confirm the adequacy of the models to the behavior of the respective systems in the failure flow (the experiments involved comparing the behavior of the models with the expected behavior in the failure flow of the systems for which they were built, using either the sets of all possible system state vectors or their random subsets). An example is given to demonstrate the application of the proposed method. It is shown that the obtained GL-model fully corresponds to the behavior of the system for which it was built.

Further work can be devoted to the study of the possibilities of further modification of the obtained models when analyzing the occurrence of dangerous states during the operation of the FTMS.

REFERENCES

1. 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.

2. 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*. 2021; 5 (1): 25–34. DOI: https://doi.org/10.15276/aait.05.2022.2.

3. 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.

4. 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.

5. Drozd, O., Ivanova, O., Zashcholkin, K., Romankevich, V. & Drozd, Yu. "Checkability Important for Fail-Safety of FPGA-based Components in Critical Systems". *CEUR Workshop Proceedings*. 2021; 2853: 471–480, https://www.scopus.com/record/display.uri?eid=2-s2.0-85104838273&origin=resultslist.

6. 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 2023 – Proceedings.* 2023. p. 223–226, https://www.scopus.com/record/display.uri?eid=2-s2.0-85167870342&origin=resultslist. DOI: https://doi.org/10.1109/ICEST58410.2023.10187275.

7. Romankevich, A., Feseniuk, A., Romankevich, V. & Sapsai, T. "About a fault-tolerant multiprocessor control system in a pre-dangerous state". *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*. 2018. p. 207–211, https://www.scopus.com/record/display.uri?eid=2-s2.0-85050654902&origin=resultslist. DOI: https://doi.org/10.1109/DESSERT.2018.8409129.

8. Safari, S. et al. "A Survey of Fault-Tolerance Techniques for Embedded Systems From the Perspective of Power, Energy, and Thermal Issues". *IEEE Access*. 2022: 10: 12229–12251, https://www.scopus.com/record/display.uri?eid=2-s2.0-85123365684&origin=resultslist. DOI: https://doi.org/10.1109/ACCESS.2022.3144217.

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/record/ display.uri?eid=2-s2.0-85090902832&origin=resultslist. DOI: https://doi.org/10.3390/electronics9091513.

10. Hu, Q., Niu, G. & Wang, C. "Spacecraft attitude fault-tolerant control based on iterative learning observer and control allocation". *Control Allocation for Spacecraft under Actuator Faults*. 2018; 75: 245–253, https://www.scopus.com/record/display.uri?eid=2-s2.0-85041462830& origin=resultslist. DOI: https://doi.org/10.1016/j.ast.2017.12.031.

11. 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/record/ display.uri?eid=2-s2.0-85132157930&origin=resultslist. DOI: https://doi.org/10.1016/j.ifacol.2022.04.134.

12. Dumitrescu, M. "Fault tolerant control multiprocessor systems modelling using advanced stochastic petri nets". *Procedia Technology*. 2016; 22: 623–628. DOI: https://doi.org/10.1016/j.protcy.2016.01.129.

13. Wang, Z. et al. "Research on Joint Optimal Scheduling of Task Energy Efficiency and Reliability in Heterogeneous Multiprocessor Real-time System". *2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA)*. 2022. p. 17–22, https://www.scopus.com/record/display.uri?eid=2-s2.0-85127423511&origin=resultslist. DOI: https://doi.org/10.1109/ICPECA53709.2022.9719271.

14. Canal, R. et al. "Predictive Reliability and Fault Management in Exascale Systems: State of the Art and Perspectives". *ACM Computing Surveys*. 2020; 53 (5): 1–32. DOI: https://doi.org/10.1145/3403956.

15. 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.

16. Kuo, W. & Zuo, M. “Optimal Reliability Modeling: Principles and Applications”. *John Wiley & Sons*. 2003.

17. Romankevich, V. A., Morozov, K. V., Feseniuk A. P., Romankevich, A. M. & Zacharioudakis, L. “On evaluation of reliability increase in fault-tolerant multiprocessor systems”. *Applied Aspects of Information Technology*. 2024; 7 (1): 81–95. DOI: https://doi.org/10.15276/ aait.07.2024.7.

18. Huang, L. & Qiang, X. “On modeling the lifetime reliability of homogeneous manycore systems”. *14th IEEE Pacific Rim International Symposium on Dependable Computing*. 2008. p. 87–94, https://www.scopus.com/record/display.uri?eid=2-s2.0-60349106919&origin=resultslist. DOI: https://doi.org/10.1109/PRDC.2008.23.

19. Zimmermann, A. “Reliability modeling and evaluation of dynamic systems with stochastic Petri nets (Tutorial)”. *VALUETOOLS 2013. 7th International Conference on Performance Evaluation Methodologies and Tools*. 2014. p. 324264. DOI: https://doi.org/10.4108/icst.valuetools.2013.254370.

20. 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/record/display.uri?eid=2-s2.0-84928741612&origin=resultslist. DOI: https://doi.org/10.1016/j.ress.2015.04.005.

21. Xing, L. “Reliability modeling and analysis of complex hierarchical systems”. *International Journal of Reliability, Quality and Safety Engineering*. 2005; 12 (6): 477–492, https://www.scopus.com/record/display.uri?eid=2-s2.0-29144462463&origin=resultslist. DOI: https://doi.org/10.1142/S0218539305001963.

22. Chao, M. T. & Lin, G. D. “Economical design of large consecutive k-out-of-n:F system”. *IEEE Transaction on Reliability*. 1984; R-33 (1): 411–413, https://www.scopus.com/record/ display.uri?eid=2-s2.0-0021579839&origin=resultslist. DOI: https://doi.org/10.1109/TR.1984.5221883.

23. Barlow, R. E. & Heidtmann K. D. “Computing k-out-of-n System Reliability”. *IEEE Transactions on Reliability*. 1984; R-33 (4): 322–323, https://www.scopus.com/record/display.uri?eid=2-s2.0-0021506074& origin=resultslist. DOI: https://doi.org/10.1109/TR.1984.5221843.

24. Huang, Y., Lin, L., Xu, L. & Hsieh, S.-Y. “Probabilistic Reliability via Subsystem Structures of Arrangement Graph Networks”. *IEEE Transactions on Reliability*. 2024; 73 (1): 279–289, https://www.scopus.com/record/display.uri?eid=2-s2.0-85168268383&origin=resultslist. DOI: https://doi.org/10.1109/TR.2023.3301629.

25. Rushdi, A. M. “Utilization of symmetric switching functions in the computation of k out-of-a system reliability”. *Microelectronics and Reliability*. 1986; 26 (5): 973–987, https://www.scopus.com/record/display.uri?eid=2-s2.0-0022865215&origin=resultslist. DOI: https://doi.org/10.1016/0026-2714(86)90239-8.

26. Belfore, L. A. “An $O(n\cdot(\log_2(n))^2)$ algorithm for computing the reliability of k-out-of-n:G and k-to-l-out-of-n:G systems”. *IEEE Transactions on Reliability*. 1995; 44 (1): 132–136, https://www.scopus.com/record/display.uri?eid=2-s2.0-0029274422&origin=resultslist. DOI: https://doi.org/10.1109/24.376535.

27. 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/record/display.uri?eid=2-s2.0-85163222383&origin=resultslist. DOI: https://doi.org/10.1002/net.22168.

28. Gökdere, G., Gürcan, M. & Kılıç, M. “A new method for computing the reliability of consecutive k-out-of-n:F systems”. *Open Physics*. 2016; 14 (1): 166–170, https://www.scopus.com/record/display.uri?eid=2-s2.0-84973539438&origin=resultslist. DOI: https://doi.org/10.1515/phys-2016-0015.

29. Belaloui, S. & Bennour, B. “Reliability of linear and circular consecutive-k-out-of-n systems with

shock model". *Afrika Statistika*. 2015; 10 (1): 795–805. DOI: http://doi.org/10.16929/as/2015.795.70.

30. 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*. 2022; 224: 108549, https://www.scopus.com/record/display.uri?eid=2-s2.0-85142023916&origin=resultslist. DOI: https://doi.org/10.1177/1748006X221130540.

31. 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/record/display.uri?eid=2-s2.0-85100736521&origin=resultslist.
DOI: https://doi.org/10.1016/j.ress.2021.107532.

32. 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.

33. Yi, H., Cui, L. & Gao, H. "Reliabilities of Some Multistate Consecutive k Systems". *IEEE Transactions on Reliability*. 2020; 69 (2): 414–429, https://www.scopus.com/record/display.uri?eid=2-s2.0-85074664125&origin=resultslist. DOI: https://doi.org/10.1109/TR.2019.2897726.

34. Gökdere, G. & Gürcan, M. "Dynamic Reliability Evaluation of Linear Consecutive k-out-of-n0:F System With Multi-State Components". *ITM Web of Conferences*. 2018; 22: 01057.
DOI: https://doi.org/10.1051/itmconf/20182201057.

35. Eryilmaz, S. & Kan, C. "Dynamic Reliability Evaluation of Consecutive-k-Within-m-Out-of-n:F System". *Communications in Statistics: Simulation and Computation*. 2011; 40 (1): 58–71, https://www.scopus.com/record/display.uri?eid=2-s2.0-80052953276&origin=resultslist.
DOI: https://doi.org/10.1080/03610918.2010.530366.

36. 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/record/display.uri?eid=2-s2.0-84937631892&origin=resultslist.
DOI: https://doi.org/10.1016/j.apm.2014.12.042.

37. Eryilmaz, S., Kan, C. & Akici, F. "Consecutive k-within-m-out-of-n:F system with exchangeable components". *MPRA Paper, University Library of Munich, Germany*. 2009, https://www.scopus.com/record/display.uri?eid=2-s2.0-68949105668&origin=resultslist.
DOI: https://doi.org/10.1002/nav.20354.

38. Levitin, G. "Consecutive k-out-of-r-from-n system with multiple failure criteria". *IEEE Transactions on Reliability*. 2004; 53 (3): 394–400, https://www.scopus.com/record/display.uri?eid=2-s2.0-4544231384&origin=resultslist. DOI: https://doi.org/10.1109/TR.2004.833313.

39. 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/record/display.uri?eid=2-s2.0-85069848232&origin=resultslist. DOI: https://doi.org/10.1142/S0218539320500035.

40. Triantafyllou, I. "m-Consecutive-k-out-of-n: F Structures with a Single Change Point". *Mathematics*. 2020; 8 (12): 2203, https://www.scopus.com/record/display.uri?eid=2-s2.0-85097534209&origin=resultslist. DOI: https://doi.org/10.3390/math8122203.

41. 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/record/display.uri?eid=2-s2.0-85100161580&origin=resultslist.
DOI: https://doi.org/10.21123/bsj.2021.18.2.0430.

42. Triantafyllou, I. S. "Combined m-Consecutive-k-Out-of-n: F and Consecutive kc-Out-of-n:F Structures with Cold Standby Redundancy". *Mathematics*. 2023; 11 (12): 1–13, https://www.scopus.com/record/display.uri?eid=2-s2.0-85164204439&origin=resultslist.
DOI: https://doi.org/10.3390/math11122597.

43. Higashiyama, Y., Cai, X. & Rumchev, V. "An O(n) Algorithm to Compute the Reliability of Consecutive k-out-of-r-from-N :F System under the Condition of r < 2k". *The 14th World Mult-Conference on Systemics, Cybernetics and Informatics ((WMSCI 2010)*. 2010. p. 151–154.

https://www.scopus.com/record/display.uri?eid=2-s2.0-84870210090&origin=resultslist.

44. Kamalja, K. K. & Shinde, R. L. "On the Reliability of (n, f, k) and <n, f, k> Systems". *Communications in Statistics – Theory and Methods*. 2014; 43 (8): 1649–1665, https://www.scopus.com/record/display.uri?eid=2-s2.0-84898856152&origin=resultslist. DOI: https://doi.org/10.1080/03610926.2012.673674.

45. Cui, L. R, Kuo, W., Li, J. L. & Xie, M. "On the dual reliability systems of (n,f,k) and <n,f,k>". *Statistics & Probability Letters*. 2006; 76 (11): 1081–1088, https://www.scopus.com/record/display.uri?eid=2-s2.0-33646113833&origin=resultslist. DOI: https://doi.org/10.1016/j.spl.2005.12.004.

46. Triantafyllou, I. & Koutras, M. "Reliability Properties of (n,f,k) Systems". *IEEE Transactions on Reliability*. 2014; 63 (1): 357–366, https://www.scopus.com/record/display.uri?eid=2-s2.0-84896314614&origin=resultslist. DOI: https://doi.org/10.1109/TR.2014.2299495.

47. Makri, F. S. "On circular m-consecutive-k,l-out-of-n:F systems". *IOP Conference Series: Materials Science and Engineering*. 2017; 351: 012005, https://www.scopus.com/record/display.uri?eid=2-s2.0-85050693770&origin=resultslist. DOI: https://doi.org/10.1088/1757-899X/351/1/012005.

48. 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/record/display.uri?eid=2-s2.0-85019236438&origin=resultslist. DOI: https://doi.org/10.1016/j.ress.2017.05.023.

49. Yamamoto, H. & Akiba, T. "A recursive algorithm for the reliability of a circular connected-(r,s)-out-of-(m,n):F lattice system". *Computers & Industrial Engineering*. 2005; 49 (1): 21–34, https://www.scopus.com/record/display.uri?eid=2-s2.0-23144441315&origin=resultslist. DOI: https://doi.org/10.1016/j.cie.2005.01.015.

50. Cui, L., Lin, C. & Du, S. "m-Consecutive-k,l-Out-of-n Systems". *IEEE Transactions on Reliability*. 2015; 64 (1): 386–393,. https://www.scopus.com/record/display.uri?eid=2-s2.0-85027947144&origin= resultslist. DOI: https://doi.org/10.1109/TR.2014.2337091.

51. 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/record/display.uri?eid=2-s2.0-85129557917&origin=resultslist. DOI: https://doi.org/10.1016/j.ress.2022.108549.

52. Zhao, X., Cui, L. R., Zhao, W. & Liu, F. "Exact reliability of a linear connected-(r, s)-out-of-(m, n):F system". *IEEE Transactions on Reliability*. 2011; 60 (3): 689–698, https://www.scopus.com/record/display.uri?eid=2-s2.0-80052411927&origin=resultslist. DOI: https://doi.org/10.1109/TR.2011.2139770.

53. 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/record/display.uri?eid=2-s2.0-85148634528&origin=resultslist. DOI: https://doi.org/10.1007/s11009-023-09984-3.

54. Lin, C., Cui L. R., Coit D. W. & Lv, M. "Reliability modeling on consecutive-kr-out-of-nr:F linear zigzag structure and circular polygon structure". *IEEE Transactions on Reliability*. 2016; 65 (3): 1509–1521, https://www.scopus.com/record/display.uri?eid=2-s2.0-84973885922&origin=resultslist. DOI: https://doi.org/10.1109/TR.2016.2570545.

55. Lee, W. S., Grosh, D. L., Tillman, F. A. & Lie, C. H. "Fault tree analysis, methods and applications:a review".IEEE Transactions on Reliability. 1985; 34 (3): 194–203. DOI: https://doi.org/10.1109/TR.1985.5222114.

56. 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/record/display.uri?eid=2-s2.0-85047465084&origin=resultslist. DOI: https://doi.org/10.1007/978-3-319-91008-6_19.

57. Romankevich, A. M., Karachun, L. F. & Romankevich, V. A. "Graph-logical models for the analysis

of complex fault-tolerant computing systems” (in Russian). *Electronic Modeling*. 2001; 23 (1): 102–111.

58. Romankevich, A. M., Romankevich, V. A., Kononova, A. A. & Rabah Al Shbul “On some features of GL-models K(2, n)” (In Russian). *Visnyk NTUU “KPI” – Informatics, Operation and Computer Science*. 2004; 41: 85–92.

59. Romankevich, V. A., Potapova, E. R., Bakhtari Kh. & Nazarenko, V. V. “GL-model of behavior of fault-tolerant multiprocessor systems with a minimal number of lost edges” (In Russian). *Visnyk NTUU “KPI” – Informatics, Operation and Computer Science*. 2006; 45: 93–100.

60. Romankevitch, A. M., Morozov, K. V., Mykytenko, S. S. & Kovalenko, O. P. “On the cascade GL-model and its properties”. *Applied Aspects of Information Technology*. 2022; 5 (3): 256–271. DOI: https://doi.org/10.15276/aait.05.2022.18.

61. Romankevitch, A., Morozov, K., Romankevich, V., Halytskyi, D. & Zacharioudakis. E. “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/record/display.uri?eid=2-s2.0-85169044963&origin=resultslist.
DOI: https://doi.org/10.1007/978-3-031-36118-0_21.

62. Romankevich, V. A. & Kononova A. A. “On one method for converting GL-models of behavior of fault-tolerant multiprocessor systems in a failure flow” (in Russian). *Radio-Electronic and Computer Systems*. 2007; 7: 49–56.

63. Romankevich, V. A., Morozov, K. V. & Feseniuk, A. P. “On one method of modification of edge functions of GL-models” (in Russian). *Radio-Electronic and Computer Systems*. 2014; 6: 95–99.

64. Romankevich, A., Maidaniuk, I., Feseniuk, A. & Romankevich, V. “Complexity Estimation of GL-Models for Calculation FTMS Reliability”. *Advances in Intelligent Systems and Computing*. 2020; 938: 369–377, https://www.scopus.com/record/display.uri?eid=2-s2.0-85064536987&origin=resultslist.
DOI: https://doi.org/10.1007/978-3-030-16621-2_34.

65. Maidaniuk, I. V., Morozov, K. V., Potapova, E. R. & Shuriga, A. V. “On one property of the GL-model with a minimal number of lost edges” (In Russian). *Naukovy Visnyk of Chernivtsi University. Series: Computer Systems and Components*. 2010; 1 (2): 31–34.

Conflicts of Interest: the authors declare no conflict of interest

Received 27.02.2024
Received after revision 30.04.2024
Accepted 14.05.2024

DOI: https://doi.org/10.15276/aait.07.2024.13
УДК 004.05

Про метод побудови небазових GL-моделей на основі комбінації реберних функцій базових моделей

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
Морозова Анна Володимирівна [1)]
ORCID: https://orcid.org/0009-0004-6738-8799; a.vl.morozova@gmail.com
Лефтеріс Захаріудакіс [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

Робота присвячена проблемі побудови GL-моделей поведінки в потоці відмов небазових відмовостійких багатопроцесорних систем. Такі моделі можуть використовуватися для розрахунку параметрів надійності останніх. Система в залежності від виконання певних умов є стійкою до відмов різної кратності. Ці умови залежать лише від станів процесорів системи та можуть бути представлені спеціальними булевими виразами. Запропоновано спосіб побудови GL-моделей таких систем, який базується на комбінуванні виразів реберних функцій допоміжних базових моделей. При цьому, особливістю побудованих запропонованим способом моделей є те, що вони базуються на циклічних графах. Це спрощує процес оцінки їх зв'язності, а також спрощує аналіз роботи моделі. Крім того, це дозволяє за необхідності додатково використовувати інші методи модифікації GL-моделей. Спосіб передбачає використання допоміжних моделей, що мають однакову кількість ребер. З метою вирівнювання кількості ребер допоміжні моделі можуть бути доповнені додатковими ребрами із реберними функціями спеціального вигляду. Показано, що таке доповнення не змінює поведінки моделей. Окрім того, описано процедуру ортогоналізації виразів умов, що має проводитися, якщо умови можуть виконуватися одночасно. Показано, що вирази реберних функцій отриманих GL-моделей, котрі можуть бути досить складними, іноді можна суттєво спростити. Проведені численні експерименти, що підтверджують адекватність моделей (побудованих запропонованим способом) поведінці відповідних систем в потоці відмов. Наведено приклад, що демонструє застосування запропонованого способу. Проведено аналіз отриманої моделі та показано, що вона відповідає поведінці системи, для якої її було побудовано.

Ключові слова: відмовостійкі багатопроцесорні системи; оцінка надійності; GL-моделі; системи *k*-з-*n*

ABOUT THE AUTHORS

Vitaliy A. Romankevich - Doctor of Engineering Sciences, Professor, Head of System Programming and Special 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 та спеціальних комп'ютерних систем. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна

Kostiantyn V. Morozov - PhD, Assistant of System Programming and Special 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, асистент кафедри Системного програмування та спеціальних 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, Professor of System Programming and Special 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, професор кафедри Системного програмування та спеціальних комп'ютерних систем. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна

Anna V. Morozova - student, System Programming and Special Computer System Department, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Peremogy Ave. Kyiv, 03056, Ukraine
ORCID: https://orcid.org/0009-0004-6738-8799; a.vl.morozova@gmail.com
Research field: GL-models; Fault-tolerant multiprocessor systems reliability estimation; Discrete mathematics

Морозова Анна Володимирівна - студентка, кафедра Системного програмування та спеціальних комп'ютерних систем. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна

Lefteris Zacharioudakis - PhD, 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

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