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

On the modification of GL-models by adding edges to a cyclic graph

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 suggests a method for constructing GL-models of fault-tolerant multiprocessor systems. These models can be used, in particular, to estimate the reliability parameters of the latter by conducting statistical experiments with models of their behavior in the failure flow. Two cases are considered: the non-basic system, unlike the basic system, is resistant to some failures of increased multiplicity, or else, on the contrary, the non-basic system is vulnerable to certain failures that do not lead to the failure of the basic system. In this case, the condition under which the system's behavior differs from the baseline corresponds to a Boolean expression, that depends on the values of the elements of the system state vector, which characterizes the states of its processors in the failure flow. According to the method proposed in the article, a model of such system is built by adding an edge or several edges to the so-called MLE-model, a type of GL-model, that can be constructed for any basic system and is based on cyclic graphs. The edge function for this edge is formed based on the aforementioned Boolean expression. The models constructed by the proposed method are also based on cyclic graphs, which, in particular, significantly simplify the procedure for assessing the connectivity of the last ones. A series of experiments have been conducted to confirm the adequacy of the models (obtained by the proposed method) to the behavior of systems in the failure flow. This work presents examples that demonstrate the process of constructing GL-models for non-basic fault-tolerant multiprocessor systems using the proposed method for both of the above cases.

Keywords: GL-models; MLE-models; non-basic fault-tolerant multiprocessor systems

***For citation*: Romankevich, V. A., Morozov, K. V., Romankevich, A. M., Morozova A. V. & Zacharioudakis, L. "On the modification of GL-models by adding edges to a cyclic graph". *Herald of Advanced Information Technology*. 2024; Vol. 7 No. 2: 185–198. DOI: https://doi.org/10.15276/hait.07.2024.13**

INTRODUCTION

In the modern world, various automated and automatic systems are becoming increasingly widespread. Their use allows, on the one hand, to free a person from the need to perform monotonous work, and, on the other hand, to reduce the possible negative impact of the human factor. One of the key components of such systems is their control system (CS) [1, 2]. It receives data from sensors and control devices and, on their basis, generates control signals for the system's executive units. Obviously, the correct functioning of the system as a whole becomes impossible in the event of a failure of its CS.

For some systems, in particularly the so-called critical application systems (CAS) [3, 4], [5], whose failure may lead to significant material losses, threaten human life and health, etc., increased reliability requirements may be imposed. Accordingly, their CS must also be highly reliable. Therefore, it is convenient to perform them on the basis of the so-called fault-tolerant multiprocessor systems (FTMS) [6, 7], [8, 9], [10, 11], [12], consisting of several processors and being resistant to the failure of some of them. In this way, high levels of both reliability and performance of these systems can be achieved, which is also often important for CAS.

Sooner or later, the developer of a FTMS, faces the task of calculating its reliability parameters, for example, the probability of failure-free operation [13, 14], [15]. This task can be solved by using various methods [16, 17], [18, 19], [20], which can be generally divided into two groups [21, 22].

The methods of the first group are based on the construction of analytical expressions for calculating the corresponding parameters. Their advantage is the ability to obtain results with high accuracy, but their disadvantage is non-universality: for each type of system, it is often necessary to create a new calculation method [23, 24], [25, 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 conducting statistical experiments with models of system behavior in the failure flow [55, 56], [57]. Their advantage is that they can be used for any type of system, but their disadvantage is that parameter estimation can usually be performed only with a certain level of accuracy, which generally depends on the number of experiments performed.

The so-called GL-models, which combine the properties of graphs and Boolean functions, can be used as models of the behavior of FTMS in the failure flow [56, 57]. The model is based on an undirected graph, where each edge corresponds to a Boolean edge function that depends on the elements of the system state vector (Fig. 1). Each element of this vector corresponds to the state in the failure flow of the particular system component (1 – operational, 0 – failed). If the edge function takes a zero value, the corresponding edge is excluded from the graph.

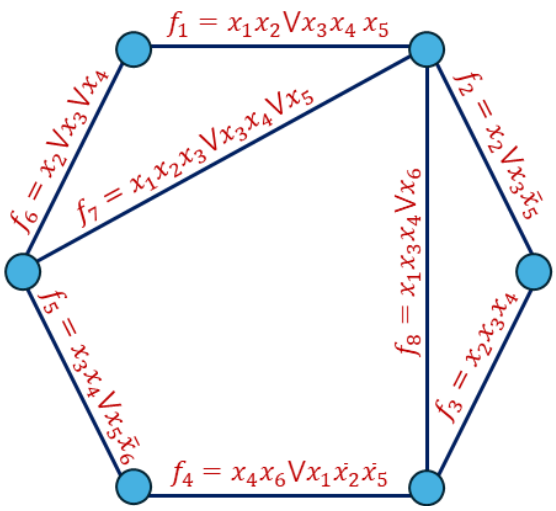

Fig. 1. **Example of GL-model**
***Source:* compiled by the authors**

The connectivity of the model's graph corresponds to the state of the system in the failure flow: connected – an operational system, loss of connectivity of the graph – system failure.

There are various methods for building GL-models [57, 58], [59, 60], [61], but they usually allow building models of so-called basic systems (basic models), those are resistant to any failures, whose multiplicity does not exceed a certain threshold value. The basic system, consisting of n processors and resilient to the failure of no more than m of them and its corresponding model are denoted by $K(m, n)$. Basic GL-models are often based on cyclic graphs, which significantly simplify the process of assessing their connectivity.

Real FTMS, especially CS, can be resistant to some failures of a certain multiplicity and, at the same time, unstable to other failures of the same multiplicity. Such systems, as well as their corresponding models, are called non-basic.

FORMULATION OF THE PROBLEM

The non-basic GL-models are usually built by modifying basic ones. This can be done either by changing the structure of the graph (drawing additional edges) [62, 63] or by changing the expressions of their edge functions [64, 65]. The approaches of the first group can lead to the loss of the cyclic structure of the graph, which complicates the process of assessing the connectivity of the last one, and can also lead to other side effects [65]. The approaches of the second group do not have this drawback but have not been fully investigated.

The problem of creating approaches that will allow building GL-models for non-basic FTMS without violating the cyclic structure of their graph is relevant.

THE METHOD OF BUILDING NON-BASIC GL-MODELS

Among the non-basic FTMS, it is worth highlighting those whose behavior in the failure flow is quite close to the behavior of some basic $K(m, n)$ system, and differs from it only in some cases. From a practical point of view, it is advisable to consider the following two situations.

1. Fault-tolerant multiprocessor systems, unlike the basic system, are also resistant to some failures of multiplicity $m + 1$.

2. Fault-tolerant multiprocessor systems, unlike the basic system, is vulnerable to some failures of multiplicity m.

We will not consider other situations, such as fault tolerance of higher multiplicity or fault tolerance of lower multiplicity. Also, we will not consider the situation when the system is resistant to some failures of multiplicity $m + 1$ and at the same time vulnerable to some failures of multiplicity m.

We will also assume that the condition (denoted as *C*), under which the system's behavior differs from the basic one, can be represented by a Boolean expression (denoted as $c(X)$) that depends on the elements of the system's state vector *X*, and takes the value of 1 if the condition is met and 0 otherwise. For example, if the system is resistant to all failures of multiplicity no more than *m*, as well as to failures of multiplicity *m* + 1 if the 2nd or both the 4th and 6th processor is functional then the expression $\mathrm{c}(X) = x_2 \vee x_4 x_6$, where $X = \langle x_1, x_2, \ldots, x_n \rangle$ is the system state vector, will correspond to this condition.

We demonstrate that both of the above situations can be represented in a single way. So, in the first case, we can say that for those vectors *X*, for which $c(X) = 1$, the system's behavior corresponds to the $K(m + 1, n)$ model, and for the remaining vectors (i.e, if $c(X) = 0$) to the $K(m, n)$ model. In the second case, if $c(X) = 1$, the system's behavior corresponds to the $K(m - 1, n)$ model, and if $c(X) = 0$, to the $K(m, n)$ model. By using the inverted expression of the condition $\bar{c}(X)$ we get: the system's behavior corresponds to the $K(m, n)$ model if $\bar{c}(X) = 1$ or to the model $K(m - 1, n)$ if $\bar{c}(X) = 0$. Therefore, a non-basic FTMS, which is close to the basic $K(m, n)$ system and differs from it only by being unstable to some failures of multiplicity *m*, whose condition of occurrence is satisfied by the Boolean expression $c(X)$, can be represented as being close to the basic $K(m - 1, n)$ system and differs from it only by being stable to some failures of multiplicity *m*, whose condition of occurrence is satisfied by the Boolean expression $\bar{c}(X)$.

As the basic $K(m, n)$ model, we will use the so-called MLE-model (minimum of lost edges) [59], which can be constructed for any values of *m* and *n*, where $n \geq 1$ and $1 \leq m \leq n$.

This model has several special properties:
it is based on a cyclic graph with $\varphi(m, n) = n - m + 1$ edges, and it loses exactly $\psi(m, l)$ edges on vectors with *l* zeros [66], where

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

In other words, for vectors with less than *m* zeros, the $K(m, n)$ model will not lose edges. For vectors with *m* zeros, it will lose exactly one edge, and for vectors with more than *m* zeros, it will lose two or more edges. Note that cyclic graph on which the model is based remains connected as long as it has no more than 1 edge is lost.

Assume that an edge with some edge function *f* has been added to this model, while keeping the graph cyclic. Then if $f = 1$, the model will lose the same number of edges as the original, meaning that it will behave like the $K(m, n)$ model. However, if $f = 0$, the model will lose one more edge than the original, specifically, one edge on vectors with less than *m* zeros and more than one edge on vectors with *m* or more zeros. Accordingly its graph, in this case, will remain connected only for vectors with less than *m* zeros, which corresponds to the behavior of the $K(m - 1, n)$ model. Such behavior of the obtained model allows us to apply it (if necessary, with a certain change in parameter values) to the above systems.

Thus, we can formulate a method for constructing **GL-models of non-basic systems described above**. To obtain a model of a system that contains *n* processors and is generally resistant to the failure of no more than *m* of them, and, under the condition that corresponds the Boolean expression $c(X)$ is tolerant to failure of no more than $m + 1$ of them, it is enough to built the basic MLE-model $K(m + 1, n)$ and, keeping its graph cyclic, add an additional edge with the edge function $f(X) = c(X)$. The graph of such model will contain exactly $\varphi(m + 1, n) + 1 = n - m + 1$ edges. For the case of a system with *n* processors that is generally resistant to failure of no more than *m* of them, except for situations that correspond to the conditional Boolean expression $c(X)$, when it is tolerant to failure of no more than $m - 1$ of them, it is enough to build a basic MLE-model $K(m, n)$ and, while keeping its graph cyclic, add an additional edge with the edge function $f(X) = \bar{c}(X)$. The number of edges in the graph of such model is $\varphi(m, n) + 1 = n - m + 2$.

ADDITIONAL CONDITIONS FOR SYSTEM FAILURE

Also consider the case when there is a certain condition (denoted as *S*), under which the system fails regardless of the number of operable processors. This can happen, for example, if there are some critical nodes (processors) in the system whose operability is mandatory. As in the previous case, we assume that this condition is satisfied by a certain expression $s(X)$, which depends on the values of the elements of the system state vector and which takes the value 1 if the condition is met and 0 if it is not.

We will assume that, in one way, we have succeeded in building a model that matches the behavior of the system except for the above condition, and such a model is based on a cyclic graph. In particular, it can be some basic model or a modified basic model obtained in the way described in the previous section.

We will add two additional edges to the model with the edge functions $f_1'(X) = f_2'(X) = \bar{s}(X)$, keeping the graph cyclic. On those vectors where $s(X) = 0$ (i.e., the condition is not fulfilled), these functions will take a value equal to 1. Therefore, the corresponding edges will remain in the graph, and the model will lose exactly the same number of edges as the original one. In other words, the behavior of the models will be the same. On the same vectors where $s(X) = 1$ (i.e., the condition is met), the functions f_1' and f_2' will take a zero value, which will lead to the exclusion of the corresponding edges from the graph. Therefore, the modified model will lose two more edges than the original model. This means that the modified model will indeed show the faulty state of the system on these vectors. Note that the number of additional edges can be higher, but this will only complicate the model, usually without bringing any benefit.

It is worth noting that while the proposed approach is quite simple, the addition of two additional edges does complicate the model somewhat, which may be undesirable. If the original model always loses at least one edge (not necessarily the same one) when condition *S* is met, then it is enough to add just one edge with the corresponding edge function instead of two. In this case, when the condition *S* is not met, the behavior of the original and modified models will coincide. However, when condition *S* is met, the modified model will lose one more edge. Considering that the original model loses at least one edge, the modified system will lose at least two edges, which corresponds to the inoperable state of the system.

In particular, the proposed optimization is possible when the original model is the basic $K(m, n)$ MLE-model, and the condition *S* is satisfied on vectors with at least m zeros. Indeed, on vectors with m or more zeros, the original MLE-model will lose at least one edge. Another possible case for applying the proposed optimization is an additional modification of the GL-model obtained by the method described in the previous section, if condition *C* is always satisfied/not satisfied when condition *S* is satisfied (i.e., if the model always loses an additional edge when condition *S* is satisfied).

Furthermore, it is possible not change the number of edges of the model graph at all [65]. Let us choose a set of edge functions of the model $f_i(X)$ and replace them with edge functions of the form $f_i'(X) = f_i(X)\bar{s}(X)$, where $i \in I$, and I is the set of indices of the functions that were selected for modification. Then, in cases where the condition *S* is not fulfilled (i.e., $s(X) = 0$), the equality $f_i'(X) = f_i(X)$, meaning that the behavior of the modified and original models will coincide. However, if the condition *S* is fulfilled ($s(X) = 1$), we will get $f_i'(X) = 0$.

Therefore, it is enough to modify at least two edge functions in this way so that the model, under condition *S*, shows the inoperable state of the system, and in other cases, its behavior does not differ from that of the original model (it is also possible to modify one function and add one edge).

In certain situations, it may be sufficient to modify only one edge function of the model. This becomes possible if there exists an edge *E* in the model such that it always loses at least one of its other edges, when the condition *S* is satisfied. In this case, it is enough to modify only the function corresponding to the edge *E*. Accordingly, if condition *S* is not met, the behavior of the original and modified models will coincide. However, if it is fulfilled, the modified model will lose at least 2 edges (edge *E* and at least one more), which indeed corresponds to the inoperable state of the system.

Note that the above modification sometimes allows not only to avoid complication, but also to simplify the expressions of edge functions of the model. Therefore, if possible, it is advisable to choose those functions whose modification will lead to the least complexity of the modified model.

EXAMPLES

Example 1. The system is 2-fault-tolerant, contains 8 processors, and if the 1st and 3rd processors are operable simultaneously or 4th and 5th and 7th processors are operable simultaneously, it is 3-fault-tolerant. This condition corresponds to the expression $c(X) = x_1x_3 \vee x_4x_5x_7$.

The MLE-model $K(3, 8)$, built in accordance with [59], will contain 6 edges with the following edge functions:

$$f_1 = x_1 \vee x_2 \vee x_3x_4;$$
$$f_2 = x_1x_2 \vee x_3 \vee x_4;$$
$$f_3 = (x_1 \vee x_2)(x_1x_2 \vee x_3x_4)(x_3 \vee x_4) \vee x_5x_6x_7x_8;$$
$$f_4 = x_1x_2x_3x_4 \vee (x_5 \vee x_6)(x_5x_6 \vee x_7x_8)(x_7 \vee x_8);$$
$$f_5 = x_5 \vee x_6 \vee x_7x_8;$$
$$f_6 = x_5x_6 \vee x_7 \vee x_8.$$

We will complement it with an additional edge with the function

$$f_7 = c(X) = x_1x_3 \vee x_4x_5x_7.$$

Therefore, the resulting GL-model will indeed be based on a cyclic graph with 8 – 2 + 1 = 7 edges.

The results of the experiments show that the obtained model demonstrates the system's operable state on all vectors with two zeros, as well as on the next 29 vectors with three zeros: 00011111, 00111011, 00111110, 01011011, 01011110, 01111010, 10011011,

 ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

10011110, 10100111, 10101011, 10101101, 10101110, 10110011, 10110101, 10110110, 10111001, 10111010, 10111100, 11011010, 11100011, 11100101, 11100110, 11101001, 11101010, 11101100, 11110001, 11110010, 11110100, 11111000. Note that among the vectors of length 8 containing 3 zeros, there are exactly $C_{8-2}^{3} = 20$ such vectors in which the 1st and 3rd components have the value 1 (which corresponds to the situation where the 1st and 3rd processors are in operable state), namely: **1**0**1**00111, **1**0**1**01011, **1**0**1**01101, **1**0**1**01110, **1**0**1**10011, **1**0**1**10101, **1**0**1**10110, **1**0**1**11001, **1**0**111**0**1**0, **1**0**1**11100, **1**1**1**00011, **1**1**1**00101, **1**1**1**00110, **1**1**1**01001, **1**1**1**01010, **1**1**1**01100, **1**1**1**10001, **1**1**1**10010, **1**1**1**10100, **1**1**1**11000; is exactly $C_{8-3}^{3} = 10$ such that the 4th, 5th, and 7th components are equal to 1 (which corresponds to the situation where the 4th, 5th, and 7th processors are in operable state): 000**11**1**1**1, 001**11**0**1**1, 001**11**1**1**0, 010**11**0**1**1, 010**11**1**1**0, 011**11**0**1**0, 100**11**0**1**1, 100**11**1**1**0, **1**0**111**0**1**0, 110**11**0**1**0; and exactly $C_{8-5}^{3} = 1$ such that both the 1st and 3rd, as well as the 4th, 5th, and 7th components have the value 1 (which corresponds to the situation when both of the above conditions are simultaneously met): **1**0**111**0**1**0. It is easy to see that their total number is indeed 20 + 10 – 1 = 29, and that they do indeed constitute that the set obtained from experiments with the GL-model built above. Therefore, this model is adequate to the behavior of the system for which it was built.

Example 2. The system is 3-fault-tolerant, has 9 processors, and if the 1st processor fails or both the 4th and 6th processors fail, it is only 2-fault-tolerant. This condition corresponds to the expression $c(X) = \bar{x}_1 \vee \bar{x}_4\bar{x}_6$. Let us the MLE-model *K*(3, 9), according to [59].

It will contain 7 edges with the following edge functions:

$f_1 = x_1 \vee x_2 \vee x_3;$
$f_2 = (x_1 \vee x_2)(x_1x_2 \vee x_3) \vee x_4x_5;$
$f_3 = x_1x_2x_3 \vee x_4 \vee x_5;$
$f_4 = (x_1 \vee x_2)(x_1x_2 \vee x_3)(x_1x_2x_3 \vee x_4x_5) \wedge$
$\wedge (x_4 \vee x_5) \vee x_6x_7x_8x_9;$
$f_5 = x_1x_2x_3x_4x_5 \vee (x_6 \vee x_7)(x_6x_7 \vee x_8x_9) \wedge$
$\wedge (x_8 \vee x_9);$
$f_6 = x_6 \vee x_7 \vee x_8x_9;$
$f_7 = x_6x_7 \vee x_8 \vee x_9.$

Let us add an additional edge with the function

$$f_8 = \bar{c}(X) = \overline{\bar{x}_1 \vee \bar{x}_4\bar{x}_6} = x_1\overline{\bar{x}_4\bar{x}_6} = x_1(x_4 \vee x_6).$$

The resulting model will be based on a cyclic graph with 9 – 3 + 2 = 8 edges.

According to the results of the experiments, the model corresponded to the behavior of the 3-fault-tolerant system everywhere except for the next 34 vectors with three zeros, where the model showed the inoperable state of the system: 011111100, 011111010, 111010110, 011110110, 011101110, 011011110, 010111110, 001111110, 011111001, 111010101, 011110101, 011101101, 011011101, 010111101, 001111101, 111010011, 011110011, 011101011, 011011011, 010111011, 001111011, 111000111, 011100111, 110010111, 101010111, 011010111, 010110111, 001110111, 011001111, 010101111, 001101111, 010011111, 001011111, 000111111.

Note that among the vectors of length 9 with 3 zeros, there are exactly $C_{9-1}^{3-1} = 28$ such vectors in which the 1st element is 0 (which corresponds to the fault of the 1st processor), namely: **0**11111100, **0**11111010, **0**11110110, **0**11101110, **0**11011110, **0**10111110, **0**01111110, **0**11111001, **0**11110101, **0**11101101, **0**11011101, **0**10111101, **0**01111101, **0**11110011, **0**11101011, **0**11011011, **0**10111011, **0**01111011, **0**11100111, **0**11010111, **0**10110111, **0**01110111, **0**11001111, **0**10101111, **0**01101111, **0**10011111, **0**01011111, **0**00111111; there are exactly $C_{9-2}^{3-2} = 7$ such that the 4th and 6th elements have a zero value (which corresponds to a fault of the 4th and 6th processor): 111**0**1**0**110, 111**0**1**0**101, 111**0**1**0**011, 111**0**0**0**111, 110**0**1**0**111, 101**0**1**0**111, 011**0**1**0**111; and exactly $C_{9-3}^{3-3} = 1$ such that the 1st, 4th, and 6th elements are 0 (which corresponds to the fulfillment of both conditions simultaneously): **0**11**0**1**0**111. The total number of such vectors is indeed 28 + 7 – 1 = 34, and they indeed comprise the above set obtained in the experiments. Therefore, the built GL-model is adequate to the behavior of the system described in the example.

Example 3. The system consists of 10 processors and is 3-fault-tolerant, but fails if the 4th processor fails or if both the 1st and 2nd processors fail. The following expression corresponds to this condition $s(X) = \bar{x}_1\bar{x}_2 \vee \bar{x}_4$. Let us build the *K*(3, 10) model according to [59].

It will contain 8 edges with the following edge functions:

$f_1 = x_1 \vee x_2 \vee x_3;$
$f_2 = (x_1 \vee x_2)(x_1x_2 \vee x_3) \vee x_4x_5;$
$f_3 = x_1x_2x_3 \vee x_4 \vee x_5;$
$f_4 = (x_1 \vee x_2)(x_1x_2 \vee x_3)(x_1x_2x_3 \vee x_4x_5) \wedge$
$\wedge (x_4 \vee x_5) \vee x_6x_7x_8x_9x_{10};$
$f_5 = x_1x_2x_3x_4x_5 \vee (x_6 \vee x_7)(x_6x_7 \vee x_8) \wedge$
$\wedge (x_6x_7x_8 \vee x_9x_{10});$
$f_6 = x_6 \vee x_7 \vee x_8;$
$f_7 = (x_6 \vee x_7)(x_6x_7 \vee x_8) \vee x_9x_{10};$
$f_8 = x_6x_7x_8 \vee x_9 \vee x_{10}.$

Let us add two additional edges to the model with the same edge functions

$$f_9 = f_{10} = \bar{s}(X) = \overline{\overline{x}_1\overline{x}_2 \vee \overline{x}_4} = \overline{\overline{x}_1\overline{x}_2}x_4 = \\ = (x_1 \vee x_2)x_4.$$

Therefore, the resulting model will be based on a cyclic graph with 10 edges.

Experiments show that the model generally corresponds to the behavior of a 3-fault-tolerant system, except for 54 vectors, namely: one vector with 1 zero (1110111111), 10 vectors with 2 zeros (1110111110, 1110111101, 1110111011, 1110110111, 1110101111, 1110011111, 1100111111, 1010111111, 0110111111, 0011111111) and 43 vectors with 3 zeros (1110111100, 1110111010, 1110110110, 1110101110, 1110011110, 1100111110, 1010111110, 0110111110, 0011111110, 1110111001, 1110110101, 1110101101, 1110011101, 1100111101, 1010111101, 0110111101, 0011111101, 1110110011, 1110101011, 1110011011, 1100111011, 1010111011, 0110111011, 0011111011, 1110100111, 1110010111, 1100110111, 1010110111, 0110110111, 0011110111, 1110001111, 1100101111, 1010101111, 0110101111, 0011101111, 1100011111, 1010011111, 0110011111, 0011011111, 1000111111, 0100111111, 0010111111, 0001111111). On these vectors, the modified model shows the inoperable state of the system.

The vector with 1 zero (111**0**111111) corresponds to the situation when only the 4th processor in the system is failed. There are indeed only $C_{10-1}^{1-1} = 1$ such vectors. In this case, an additional condition is indeed fulfilled, under which the system is faulty. Note that there are no vectors with 1 zero in which both the 1st and 2nd elements have zero values (which corresponds to the failure of the 1st and 2nd processors). Therefore, the above vector is the only possible vector with 1 zero for which the additional condition for system failure is fulfilled.

Among the vectors with 2 zeros, we can identify those where the 4th and any other element is zero (which meets the condition). These are the following $C_{10-1}^{2-1} = 9$ vectors: 111**0**111110, 111**0**111101, 111**0**111011, 111**0**110111, 111**0**101111, 111**0**011111, 110**0**111111, 101**0**111111, 011**0**111111. In addition, the condition is also fulfilled when the 1st and 2nd processors are faulty, which corresponds to $C_{10-2}^{2-2} = 1$ vector **00**11111111, where both 1st and 2nd elements are equal to zero. These 10 vectors are all possible vectors for which the above condition is satisfied.

The vectors with 3 zeros include $C_{10-1}^{3-1} = 36$ vectors, where the 4th element is zero (111**0**111100, 111**0**111010, 111**0**110110, 111**0**101110, 111**0**011110, 110**0**111110, 101**0**111110, 011**0**111110, 111**0**111001, 111**0**110101, 111**0**101101, 111**0**011101, 110**0**111101, 101**0**111101, 011**0**111101, 111**0**110011, 111**0**101011, 111**0**011011, 110**0**111011, 101**0**111011, 011**0**111011, 111**0**100111, 111**0**010111, 110**0**110111, 101**0**110111, 011**0**110111, 111**0**001111, 110**0**101111, 101**0**101111, 011**0**101111, 110**0**011111, 101**0**011111, 011**0**011111, 100**0**111111, 010**0**111111, 001**0**111111), and also $C_{10-2}^{3-2} = 8$ such that the 1st and 2nd elements have zero values (**00**11111110, **00**11111101, **00**11111011, **00**11110111, **00**11101111, **00**11011111, **00**10111111, **00**01111111). There is also exactly $C_{10-3}^{3-3} = 1$ vector with 3 zeros, which is included in both of the above sets. This is a vector in which the 4th, 1st and 2nd elements have zero values: **00**1**0**111111. The total number of vectors is indeed 36 + 8 – 1 = 43, and these are all possible vectors with 3 zeros for which the additional condition of system failure is fulfilled.

The modification can be done in another way. For example, we can add only one edge to the model with the edge function $f_9 = (x_1 \vee x_2)x_4$ and modify the expression of one of the edge functions of the original model.

For example, we will modify the edge function f_1:

$$f_1' = f_1\bar{s} = (x_1 \vee x_2 \vee x_3)\overline{\overline{x}_1\overline{x}_2 \vee \overline{x}_4} = \\ = (x_1 \vee x_2 \vee x_3)(x_1 \vee x_2)x_4 = (x_1 \vee x_2)x_4.$$

The modified model will be based on a cyclic graph with 9 edges, which will correspond to the edge functions f_1', f_2, f_3, f_4, f_5, f_6, f_7, f_8 and f_9. The behavior of such a model will be the same as that of the model with 2 edges added.

It is even possible not to add additional edges to the model at all, but modify one more edge function. For example, we will modify the expression of the edge function f_2:

$$f_2' = f_2\bar{s} = \big((x_1 \vee x_2)(x_1x_2 \vee x_3) \vee x_4x_5\big) \wedge \\ \wedge \overline{\overline{x}_1\overline{x}_2 \vee \overline{x}_4} = \big((x_1 \vee x_2)(x_1x_2 \vee x_3) \vee x_4x_5\big) \wedge \\ \wedge (x_1 \vee x_2)x_4 = (x_1 \vee x_2)(x_1x_2 \vee x_3)x_4 \vee \\ \vee (x_1 \vee x_2)x_4x_5 = (x_1 \vee x_2)(x_1x_2 \vee x_3 \vee x_5)x_4.$$

In this case, the modified model will be based on a cyclic graph with 8 edges, which will correspond to the edge functions f_1', f_2', f_3, f_4, f_5, f_6, f_7 and f_8. The behavior of this model will be the same as the previous two.

Note that the above modifications did not increase the complexity of the expressions of the model's edge functions. However, if, for example, the edge function f_6, had been chosen for modification, then although the behavior of the resulting model would have been the same as that of the previous model, the complexity of the expressions of its edge functions would have been slightly higher. The modified edge function would then have the form:

$f_6' = f_6\bar{s} = (x_6 \vee x_7 \vee x_8)\overline{\bar{x}_1\bar{x}_2 \vee \bar{x}_4} =$
$= (x_6 \vee x_7 \vee x_8)(x_1 \vee x_2)x_4.$

Therefore, we can note that each of the modified GL-models is adequate to the behavior of the system under consideration.

Example 4. The system is 2-fault-tolerant, consists of 10 processors, but fails if at least one of the next three processors is faulty: 1st, 2nd, 3rd, and at the same time, at least one of the following four processors is also faulty: 5th, 6th, 7th, and 8th. This condition is satisfied by the expression $s(X) = (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)(\bar{x}_5 \vee \bar{x}_6 \vee \bar{x}_7 \vee \bar{x}_8)$. Let us build the $K(2, 10)$ model according to [59]. This model will be based on a cyclic graph with 9 edges and the following edge functions:

$f_1 = x_1 \vee x_2;$
$f_2 = x_1x_2 \vee x_3;$
$f_3 = x_1x_2x_3 \vee x_4x_5;$
$f_4 = x_4 \vee x_5;$
$f_5 = x_1x_2x_3x_4x_5 \vee x_6x_7x_8x_9x_{10};$
$f_6 = x_6 \vee x_7;$
$f_7 = x_6x_7 \vee x_8;$
$f_8 = x_6x_7x_8 \vee x_9x_{10};$
$f_9 = x_9 \vee x_{10}.$

We can modify the model by adding 2 additional edges with edge functions

$f_{10} = f_{11} = \bar{s}(X) =$
$= \overline{(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)(\bar{x}_5 \vee \bar{x}_6 \vee \bar{x}_7 \vee \bar{x}_8)} =$
$= \overline{\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3} \vee \overline{\bar{x}_5 \vee \bar{x}_6 \vee \bar{x}_7 \vee \bar{x}_8} =$
$= x_1x_2x_3 \vee x_5x_6x_7x_8.$

The resulting model will be based on a cyclic graph with 11 edges.

According to the results of the experiments, the model corresponds to the basic 2-fault-tolerant system except for the following 12 vectors with 2 zeros, where the model shows the inoperable state of the system: 1101111011, 1011111011, 0111111011, 1101110111, 1011110111, 0111110111, 1101101111, 1011101111, 0111101111, 1101011111, 1011011111, 0111011111. These are indeed vectors corresponding to the inoperable state of the 1st (**0**111111011, **0**111110111, **0**111101111, **0**111011111), the 2nd (1**0**11111011, 1**0**11110111, 1**0**11101111, 1**0**11011111) and the 3rd (11**0**1111011, 11**0**1110111, 11**0**1101111, 11**0**1011111) and at the same time the 5th (1101**0**11111, 1011**0**11111, 0111**0**11111), the 6th (11011**0**1111, 10111**0**1111, 01111**0**1111), the 7th (110111**0**111, 101111**0**111, 011111**0**111) and the 8th (1101111**0**11, 1011111**0**11, 0111111**0**11) processors.

Note that the additional condition under which the system is inoperable corresponds to situations where at least 2 processors are faulty. In this case, the basic MLE-model $K(2, 10)$ loses at least one edge. Therefore, to modify the model, it is enough to add only one edge with the edge function $f_{10} = x_1x_2x_3 \vee x_5x_6x_7x_8$. Such a model will be based on a cyclic graph with 10 edges and its behavior will not differ from the previous one. In other words, they will show the same state of the system for the same input vectors.

We can also note that if the additional condition is met, the model will lose an edge, corresponding to the function f_3 or the edge corresponding to the function f_5. Therefore, instead of adding an additional edge to the model, we can modify any of its edge functions, except for f_3 and f_5.

For example, we can modify the function f_1:

$f_1' = f_1\bar{s} = (x_1 \vee x_2) \wedge$
$\wedge \overline{(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)(\bar{x}_5 \vee \bar{x}_6 \vee \bar{x}_7 \vee \bar{x}_8)} =$
$= (x_1 \vee x_2)(x_1x_2x_3 \vee x_5x_6x_7x_8) =$
$= x_1x_2x_3 \vee x_5x_6x_7x_8(x_1 \vee x_2).$

Therefore, the model will be based on a cyclic graph with 9 edges, which will correspond to the edge functions f_1', f_2, f_3, f_4, f_5, f_6, f_7, f_8 and f_9. The behavior of this model will coincide with the behavior of the previous models, but its complexity will be somewhat lower (due to both a smaller number of edges and the lower overall complexity of the expressions of the edge function).

Note that each of the built models is adequate to the behavior of the system considered in the example.

Example 5. The system is similar to the system in ***Example 2*** (i.e., it is 3-fault-tolerant, consisting of 9 processors, and in the case of failure of the 1st or simultaneously of the 4th and 6th processors, it is resistant only to 2 failures), but also fails in the case of simultaneous failure of the 1st and the 4th or the 1st and the 6th processors. This condition corresponds to the expression $s(X) = \bar{x}_1\bar{x}_4 \vee \bar{x}_1\bar{x}_6$.

Remember that the system from ***Example 2*** was modeled on a cyclic graph with 8 edges and the following edge functions:

$f_1 = x_1 \vee x_2 \vee x_3;$
$f_2 = (x_1 \vee x_2)(x_1x_2 \vee x_3) \vee x_4x_5;$
$f_3 = x_1x_2x_3 \vee x_4 \vee x_5;$
$f_4 = (x_1 \vee x_2)(x_1x_2 \vee x_3)(x_1x_2x_3 \vee x_4x_5) \wedge$
$\wedge (x_4 \vee x_5) \vee x_6x_7x_8x_9;$
$f_5 = x_1x_2x_3x_4x_5 \vee (x_6 \vee x_7)(x_6x_7 \vee x_8x_9) \wedge$
$\wedge (x_8 \vee x_9);$
$f_6 = x_6 \vee x_7 \vee x_8x_9;$
$f_7 = x_6x_7 \vee x_8 \vee x_9;$
$f_8 = x_1(x_4 \vee x_6).$

We can modify this model by adding 2 edges with edge functions

$$f_9 = f_{10} = \bar{s} = \overline{\bar{x}_1\bar{x}_4 \vee \bar{x}_1\bar{x}_6} = (\overline{\bar{x}_1\bar{x}_4})(\overline{\bar{x}_1\bar{x}_6}) = $$
$$= (x_1 \vee x_4)(x_1 \vee x_6) = x_1 \vee x_4x_6.$$

This way, we will get a model based on a cyclic graph with 10 edges.

The experimental results show that the model does indeed corresponds to the behavior of a 3-fault-tolerant system, except for 34 vectors with three zeros, as shown in ***Example 2***, and two vectors with two zeros: 011110111 and 011011111. These vectors really correspond to situations where the 1st and the 4th processors, and the 1st and the 6th processors are faulty in the system at the same time.

In addition, we can notice that for this example, the condition under which the system has a reduced degree of fault tolerance (condition *C* – the 1st or simultaneously the 4th and the 6th processors failure) is always fulfilled when an additional condition is met, under which the system loses its operability at all (condition *S* – simultaneous failure of the 1st and 4th or the 1st and the 6th processors). Due to this, to obtain the modified model, it is enough to add only one edge with the edge function $f_9 = x_1 \vee x_4x_6$. Therefore, the modified model will be based on a cyclic graph with 9 edges.

It should also be noted that each of the built models is adequate to the behavior in the failure flow of the system considered in the example.

CONCLUSIONS

The work proposes a method for constructing GL-models for non-basic fault-tolerant multiprocessor systems. Unlike basic systems, which are resilient to any failures of multiplicity no greater than some value of *m*, the systems under consideration can also be resilient to some failures of multiplicity $m + 1$ or vulnerable to some failures of multiplicity *m*. The proposed models are based on so-called MLE-models, which can be built for any basic system and are based on cyclic graphs. According to the proposed method, to obtain a model of a non-basic system, an edge with a certain function is added to the corresponding MLE-model. In this case, the model graph remains cyclic, which, in particular, simplifies the process of assessing its connectivity.

In addition to the proposed method, we consider the case when there is some additional condition under which the system becomes inoperable. In this case, it is enough to modify the GL-model by adding at least two additional edges to it.

The experiments were conducted for both the cases when the system is resilient to certain failures of multiplicity *m* + 1 and when the system is vulnerable to some failures of multiplicity *m*. These experiments involved comparing the behavior of the constructed 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. The results of these experiments confirm the correctness of the proposed method of constructing GL-models for non-basic systems. Some of these results for each case are provided in the article as examples.

Among the possible directions for future work, there is the study of possibility of combining the proposed methods with other methods of modifying GL-models. In addition, it is of interest to build GL-models of systems that are simultaneously resistant to some failures of increased multiplicity and vulnerable to certain failures of ordinary multiplicity. Another worthwhile study is the possibility of building models for cases where the system differs from the basic one more significantly (being resistant to failures of higher multiplicity or vulnerable to failures of lower multiplicity).

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". *58th International Scientific Conference on Information, Communication and Energy Systems and Technologies, ICEST. – 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". *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". *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. 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.

16. 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.

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

18. 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.

19. 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.

20. 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.

21. Kuo, W. & Zuo, M. "Optimal reliability modeling: Principles and applications". *John Wiley & Sons*. 2003.

22. 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.

23. 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.

24. 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.

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 behavior 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; 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. 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.

57. 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.

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, 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.

64. 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.

65. Morozov, K. V., Romankevich, A. M., Romankevich, V. A. “On the nature of the influence of modification of edge functions of a GL-model on its behavior in a failure flow” (In Russian). *Radio-Electronic and Computer Systems*. 2016; 6: 108–112.

66. 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 26.02.2024
Received after revision 30.04.2024
Accepted 14.05.2024

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

Про модифікацію GL-моделей шляхом додавання ребер в циклічний граф

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
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

В роботі запропоновано спосіб побудови GL-моделей відмовостійких багатопроцесорних систем. Ці моделі можуть бути використані, зокрема, для оцінки параметрів надійності останніх методом проведення статистичних експериментів із моделями їх поведінки в потоці відмов. Розглядається два випадки: небазова система, на відміну від базової, є стійкою до деяких відмов підвищеної кратності, або ж навпаки, небазова система є нестійкою до деяких відмов, котрі не призводять до виходу з ладу базової системи. При цьому, умові, за якої поведінка системи відрізняється від базової відповідає деякий булевий вираз, що залежить від значень елементів вектору стану системи, котрий характеризує стани її процесорів в потоці відмов. Відповідно до запропонованого в статті способу модель такої системи будується шляхом додавання ребра або декількох ребер до так званої МВР-моделі – одного з видів GL-моделей, котрі можуть бути побудовані для будь-яких базових систем та мають у своїй основі циклічні графи. Реберна функція для цього ребра формується на базі вищезгаданого булевого виразу. Моделі, побудовані запропонованим способом також базуються на циклічних графах, що, зокрема, суттєво спрощує процедуру оцінки зв'язності останніх. Проведено ряд експериментів, котрі підтверджують адекватність моделей (отриманих запропонованим способом) поведінці систем в потоці відмов. В роботі наведено приклади, котрі демонструють процес побудови GL-моделей для небазових відмовостійких багатопроцесорних систем запропонованим способом для обох вищезгаданих випадків.

Ключові слова: GL-моделі; МВР-моделі; небазові відмовостійкі багатопроцесорні системи

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, 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, 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, кафедра 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 (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.