DOI: https://doi.org/10.15276/aait.08.2025.9
UDC 004.05

GL-models for analyzing multiprocessor systems tolerant to one and two failures

Vitaliy A. Romankevich[1)]
ORCID: https://orcid.org/0000-0003-4696-5935; zavkaf@scs.kpi.ua. Scopus Author ID: 57193263058
Kostiantyn V. Morozov[1)]
ORCID: https://orcid.org/0000-0003-0978-6292; mcng@ukr.net. Scopus Author ID: 57222509251
Alexei M. Romankevich[1)]
ORCID: https://orcid.org/0000-0001-5634-8469; romankev@scs.kpi.ua. Scopus Author ID: 6602114176
Daniil V. Halytsky[1)]
ORCID: https://orcid.org/0009-0004-4421-3443; zipper135401@gmail.com. Scopus Author ID: 58553487600
Anton V. Zhurba[1)]
ORCID: https://orcid.org/0009-0007-3375-2590; zhurba.anton@outlook.com
[1)] National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, 37, Peremogy Ave. Kyiv, 03056, Ukraine

ABSTRACT

The work is devoted to the problem of constructing GL-models of behavior in the fault flow of non-basic fault-tolerant multiprocessor systems. We consider systems that are fault-tolerant to any one failure, and under a certain condition – to any two failures of their processors. It is assumed that this condition is satisfied by a Boolean expression that depends on the states of the system's processors. To build models of such systems, we use the previously proposed method based on combining the expressions of the edge functions of auxiliary basic GL-models corresponding to 1- and 2-fault-tolerant systems. Two alternative ways of constructing auxiliary $K(2, n)$ models are considered. It is shown that the obtained GL-models are based on the same cyclic graphs and differ only in the expressions of their edge functions. The complexity of the expressions of the edge functions of the obtained models is analyzed for the cases of using auxiliary GL-models of each type. The formulas for estimating the complexity of these expressions (the number of elementary logical operations) depending on the number of processors in the system, in particular, for cases when this number is a power of two, are obtained. Also, for the case of using one of the auxiliary models, the possibility of further simplification of the expressions of some of the edge functions is revealed. It is determined in which cases, namely, for what number of system processors, it is advisable to use auxiliary models of one or another type. Experiments confirm that the GL-models constructed in the above way adequately reflect the behavior of the system. Examples are given that demonstrate the application of the considered method of constructing non-basic GL-models and confirm the correctness of estimation the complexity of expressions of their edge functions when using auxiliary models of various types. It has also been shown that further simplification of the expressions for the edge functions in GL-models under construction is possible, particularly by reordering the edges in auxiliary models. However, this possibility has not been investigated in the general case within this article.

Keywords: fault-tolerance; multiprocessor systems; reliability estimation; GL-models; non-basic systems; cyclic graph; edge function

For citation: **Romankevich V. A., Morozov K. V., Romankevich A. M., Halytsky D. V., Zhurba A. V. “GL-models for analyzing multiprocessor systems tolerant to one and two failures”. *Applied Aspects of Information Technology.* 2025; Vol.8 No.1: 113–128. DOI: https://doi.org/10.15276/aait.08.2025.9**

INTRODUCTION

Automation of various processes and systems has become increasingly common, especially in recent decades. The control of such facilities is partially or fully relied on specialized control systems (CS) [1, 2]. This minimizes the influence of the human factor and thus avoids a number of limitations, in particular with regard to the speed and accuracy of decision-making.

The control system receives data from various sensors and produces control signals in accordance with a specialized program. The incorrect operation of the CS can obviously lead to the incorrect operation of the system as a whole, which can have negative consequences. Thus, the CS of some systems and facilities may be subject to increased reliability requirements. This is especially relevant to the so-called critical application systems (CAS) [3, 4], [5] the failure of which can lead to significant material damage, threaten human health or life, etc. (e.g., railway, aviation, space transport, military equipment, power plants, complex production facilities, etc.). It is also worth noting that often, CAS control programs are quite complex, requiring high performance from their CS.

Taking into account the above factors, it is advisable to build control systems for CAS on the contain a certain number basis of so-called fault-tolerant multiprocessor systems (FTMS), which, as their name implies, (more than one) of processor elements and remain operational even in the event of failure of some of them [6, 7], [8, 9], [10, 11], [12].

Sooner or later, the developer of a FTMS (especially if it is a CS CAS) faces the task of calculating its reliability parameters, for example, the probability of failure [13, 14], [15, 16]. Given that such FTMS may have quite complex structure, this can be a difficult task [17, 18], [19].

It is worth noting that fault-tolerant multiprocessor systems can be divided into basic and non-basic systems. A basic system contains n processors and is resilient to the failure of no more than m of them (such a system is denoted $K(m, n)$). A non-basic system may be resistant to a certain set of failures of a certain multiplicity, but unstable to other failures of the same multiplicity.

LITERATURE REVIEW

In general, we can distinguish 2 groups of methods for calculating the reliability parameters of FTMS. The methods of the first group are reduced to the formation of some analytical expressions for calculating these parameters. Their advantage is usually the accuracy and speed of calculations, but the disadvantage is non-universality. Often, new calculation methods have to be created for each type of system [20, 21], [22]. Therefore, they are convenient to use for special types of systems, especially if they can be parameterized by a small number of parameters. Such systems include, in particular, basic systems such as k-out-of-n [23, 24], [25, 26], and some non-basic systems such as k-to-l-out-of-n [26], consecutive k-out-of-n [27, 28], [29, 30], [31, 32], [33, 34], consecutive k-within-m-out-of-n [35, 36], [37], consecutive k-out-of-r-from-n [38, 39], [40] m-consecutive-k-out-of-n [41, 42], [43], (n, f, k) [44, 45], [46], $\langle n, f, k \rangle$ [44, 45], consecutive-(k, l)-out-of-n [47], m-consecutive-k,l-out-of-n [48, 49], [50], (r, s)-out-of-(m, n) [51, 52], [53], consecutive-k_r-out-of-n_r [54] etc.

The methods of the second group are based on statistical experiments with models of the behavior of the FTMS in the failure stream. Such a model takes as input a Boolean vector of the system state, each element of which characterizes the state (operable/failed) of the corresponding processor and returns the state of the system as a whole (operable/failed) [55]. As a result of a series of experiments with random vectors, it is possible to calculate the system's reliability parameters with some accuracy (which generally depends on the number of experiments conducted). The advantage of these methods is their versatility (it is enough to build a model of system behavior in the failure stream), and the disadvantage is usually higher complexity and lower accuracy of calculations.

GL-MODELS

It is convenient to use the so-called GL-models [56, 57], which combine the properties of graphs and Boolean functions, as models of the behavior of FTMS in the failure flow. A GL-model is an undirected graph, each edge of which corresponds to a Boolean function that depends on the elements of the Boolean vector of the system state (all or only some). If the function takes a zero value on a certain vector, the corresponding edge is excluded from the graph. The connectivity of the graph corresponds to the system's operability (for the processor states corresponding to the vector under consideration): a connected graph means the system is operational, an unconnected graph means the system has failed.

There are different ways to build GL-models. For example, in [58], a method for constructing models of basic $K(2, n)$ systems is proposed. Such a model is based on a cyclic graph with n edges and contains edge functions of the form

$$f_i = x_i \vee \bigwedge_{j=1}^{\left\lfloor \frac{n-1}{2} \right\rfloor} x_{(i+j) \bmod n},$$

where x_i is i-th element of the system state vector.

In turn, [59] proposes a more universal method for constructing GL-models that can be applied to any basic system of the form $K(m, n)$, where $1 \le m \le n$. According to this method, the GL-model of the $K(m, n)$ system is based on a cyclic graph with exactly $\varphi = n - m + 1$ edges. The edge functions of such a model are constructed by dividing the input vector into two parts (in general, in an arbitrary way) and constructing auxiliary GL-models for them (GL-models of the basic systems are also denoted $K(m, n)$ and called basic). The set of edge functions of the GL-model $K(m, n)$ thus consists of the following subsets.

1. The set of edge functions of the auxiliary model $K_1(m, n_1)$, built for the first part of the input vector.
2. The set of edge functions of the form

$$f_j = \kappa_1(m - i, n_1) \vee \kappa_2(i, n_2),$$

for $i = 1, 2, \ldots, m - 1$, where

$$\kappa_1(m-i,n_1)=\bigwedge_{f^{(1)}\in\Phi(K_1(m-i,n_1))} f^{(1)},$$
$$\kappa_2(i,n_2)=\bigwedge_{f^{(2)}\in\Phi(K_2(i,n_2))} f^{(2)},$$

where $\Phi(K_1(m-i,n_1))$ is the set of edge functions of the auxiliary model $K_1(m-i, n_1)$, built for the first part of the input vector, and $\Phi(K_2(i,n_2))$ is the set of edge functions of the auxiliary model $K_2(i, n_2)$, built for the first part of the input vector.the set of edge functions of the auxiliary model $K_2(m, n_2)$, built for the second part of the input vector.

Such a process of separation can continue for the auxiliary models until the models of the form $K(1, n)$ or $K(n, n)$, the shape of the edge function for which is known. Thus, the model $K(1, n)$ contains exactly n edge functions of the form $f_i = x_i$, and model $K(n, n)$ contains only one edge function of the form $f = x_1 \vee x_2 \vee \ldots \vee x_n$. If certain auxiliary models of the form $K(m, n)$ can not be constructed in principle, for example, in the case of $m > n$, the corresponding edge functions are excluded from the resulting set.

The construction of GL-models of non-basic systems (i.e., non-basic models) can be performed by modifying certain auxiliary basic models. Thus, in [60], a method for constructing a non-basic system with n processors, which, depending on the occurrence of conditions represented by Boolean expressions $c_1, c_2, \ldots, c_k$, is resistant to any failures of multiplicities not exceeding $m_1, m_2, \ldots, m_k$ respectively. To do this, it is necessary to construct auxiliary GL-models $K(m_i, n)$ for $i = 1, 2, \ldots, k$, based on cyclic graphs (for example, by the methods described in [58, 59]), after which the edge functions of the resulting model (also based on a cyclic graph) are constructed according to the expressions of the form:

$$f_i = c_1 f_i^{(1)} \vee c_2 f_i^{(2)} \vee \ldots \vee c_k f_i^{(k)}, \qquad (1)$$

where $f_i^{(j)}$ is i-th edge function of the j-th auxiliary GL-model.

In addition, GL-models, both basic and non-basic, can be used as auxiliary models for building models of more complex systems, for example, hierarchical systems consisting of several subsystems [61, 62]. It is also worth mentioning that the accuracy of calculating the reliability parameters of a FTMS using GL-models depends on the number of statistical experiments performed. Thus, reducing the complexity of conducting one experiment with the model allows to increase the accuracy of estimating the reliability parameters of FTMS while maintaining the overall complexity of calculations.

PROBLEM FORMULATION

In some cases, a FTMS (or its subsystem) may have a small degree of fault tolerance, in the simplest case equal to 1. However, the reliability of such a system may be insufficient, and therefore the developer may increase its degree of fault tolerance, for example, to the value of 2, but only for some situations (when available excess resources allow it). Thus, depending on the fulfilment of a certain condition, the system will be 1- or 2-fault-tolerant (we call it 1-/2-fault-tolerant). To build a GL-model of such a system, one can use the method described in [60], using, as suggested in [60], the $K(1, n)$ and $K(2, n)$ models built in accordance with [59] as auxiliary models. However, the method also allows the use of other GL-models that are also based on cyclic graphs, in particular, those described in [58]. It is useful for the developer of the FTMS to understand which of the approaches is appropriate in a particular case and what properties the resulting model will have.

Thus, it is relevant to investigate the properties (in particular, the complexity of expressions of their edge functions) of GL-models of non-basic 1-/2-fault-tolerant multiprocessor systems built according to [60] using as auxiliary models built according to [58] and [59].

MODELS OF SYSTEMS TOLERANT TO ONE AND TWO FAILURES

The auxiliary models in the construction of a GL-model of a 1-/2-fault-tolerant n-processor system in accordance with [60] are the $K(1, n)$ and $K(2, n)$. We will also assume that the situation when the system is 2-fault-tolerant corresponds to the Boolean expression c, i.e., when $c = 1$, the system is 2-fault-tolerant, and when $c = 0$, it is 1-fault-tolerant. The $K(1, n)$ model will be built according to [59], and it will be based on a cyclic graph with n edges and trivial edge functions of the form $f_i^{(1)} = x_i$. The model $K(2, n)$ can be constructed both according to [58] and [59].

Consider the case of building the $K(2, n)$ model according to [58]. Like the $K(1, n)$ model, it is based on a cyclic graph with n edges and edge functions of the form

$$f_i^{(2)} = x_i \vee \bigwedge_{j=1}^{\left\lfloor \frac{n-1}{2} \right\rfloor} x_{(i+j) \bmod n}.$$

The edge functions of the GL-model of a 1-/2-fault-tolerant n-processor system according to [60], considering $c_1 \equiv \bar{c}$ and $c_2 \equiv c$, will look like this:

$$f_i = c_1 f_i^{(1)} \vee c_2 f_i^{(2)} =$$
$$= \bar{c} x_i \vee c \left(x_i \vee \bigwedge_{j=1}^{\left\lfloor \frac{n-1}{2} \right\rfloor} x_{(i+j) \bmod n} \right) =$$
$$= \bar{c} x_i \vee c x_i \vee c \bigwedge_{j=1}^{\left\lfloor \frac{n-1}{2} \right\rfloor} x_{(i+j) \bmod n} =$$
$$= x_i \vee c \bigwedge_{j=1}^{\left\lfloor \frac{n-1}{2} \right\rfloor} x_{(i+j) \bmod n}.$$

Thus, the model will be based on a cyclic graph with n edges and edge functions of a fairly simple form. Note that the value of the expression c for a given vector can be calculated once (for all edge functions). Thus, it is easy to see that the complexity of calculating the value of any edge function of the resulting GL-model (excluding the complexity of calculating the value of the expression c) will be 1 disjunction and $\left\lfloor \frac{n-1}{2} \right\rfloor$ conjunctions, or in general $\left\lfloor \frac{n-1}{2} \right\rfloor + 1$ an elementary logical operation. Given that the model contains exactly n edge functions, the complexity of calculating the values of all its edge functions (excluding the complexity of calculating the value of expression c) will be n disjunctions and $n \cdot \left\lfloor \frac{n-1}{2} \right\rfloor$ conjunctions or $n \cdot \left(\left\lfloor \frac{n-1}{2} \right\rfloor + 1 \right)$ elementary logical operations.

The construction of the edge functions of the model $K(2, n)$ according to [59] is carried out according to the iterative scheme described above. Let us estimate the complexity of the expressions of the edge functions of the $K(2, n)$ model constructed in accordance with [59].To simplify this evaluation, we assume that $n = 2^t$ and that the input vector is always divided exactly in half in the auxiliary models. The complexity of expressions of edge functions will be represented as a vector $(n_c\ n_d)$, where n_c – is the number of conjunctions, and n_d – is the number of disjunctions.

According to the above iterative scheme, the set of edge functions of the model $K(2, 2^t)$ consists of the following subsets.

1. The set of edge functions of the auxiliary model $K_1(2, 2^{t-1})$, constructed for the first half of the input vector.
2. A single edge function of the form

$$f = \kappa_1(1, 2^{t-1}) \vee \kappa_2(1, 2^{t-1}),$$

where

$$\kappa_1(1, 2^{t-1}) = \bigwedge_{f^{(1)} \in \Phi\left(K_1(1, 2^{t-1})\right)} f^{(1)},$$
$$\kappa_2(1, 2^{t-1}) = \bigwedge_{f^{(2)} \in \Phi\left(K_2(1, 2^{t-1})\right)} f^{(2)},$$

where $\Phi\left(K_1(1, 2^{t-1})\right)$ is the set of edge functions of the auxiliary model $K_1(1, 2^{t-1})$, constructed for the first half of the input vector., and $\Phi\left(K_2(1, 2^{t-1})\right)$ is the set of edge functions of the auxiliary model $K_2(1, 2^{t-1})$, constructed for the second half of the input vector. Thus (taking into account the form of the edge functions of the $K(1, n)$, models, as described above), the above function has the form:

$$f = \bigwedge_{i=1}^{2^{t-1}} x_i \vee \bigwedge_{j=1}^{2^{t-1}} x_{j+2^{t-1}}$$

3. The set of edge functions of the auxiliary model $K_2(2, 2^{t-1})$, constructed for the second half of the input vector.

We denote the complexity of the edge function from paragraph 2 as $\beta(n)$. It is easy to see that

$$\beta(n) = \beta(2^t) = = \binom{n-2}{1} = \binom{2^t-2}{1}. \tag{2}$$

The complexity of the entire model $K(2, n)$ is denoted as $\alpha(n)$. It is easy to see that

$$\alpha(n) = \alpha(2^t) = 2\alpha\left(\frac{n}{2}\right) + \beta(n) = = 2\alpha(2^{t-1}) + \beta(2^t). \tag{3}$$

It should also be noted that the model $K(2, 2)$ contains a single edge function of the form $f = x_1 \vee x_2$ and, accordingly,

$$\alpha(2^1) = \beta(2^1) = \binom{0}{1}. \tag{4}$$

Using the expansion of (3) an appropriate number of times, we reduce $\alpha(2^t)$ to expression (4), substituting expression (2) for the corresponding values of β:

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

$$\alpha(2^t) = 2 \cdot \left(2 \cdot \dots \left(2\alpha(2^1) + \beta(2^2)\right) \dots + \beta(2^{t-1})\right) +$$
$$+\beta(2^t) = 2^{t-1} \cdot \alpha(2^1) + 2^{t-2} \cdot \beta(2^2) +$$
$$+2^{t-3} \cdot \beta(2^3) + \dots + 2^1 \cdot \beta(2^{t-1}) + 2^0 \cdot \beta(2^t) =$$
$$= 2^{t-1} \cdot \alpha(2^1) + \sum_{i=0}^{t-2} 2^i \cdot \beta\left(2^{t-i}\right) =$$
$$= 2^{t-1} \cdot \binom{0}{1} + \sum_{i=0}^{t-2} 2^i \cdot \binom{2^{t-i} - 2}{1} =$$
$$= \binom{0}{2^{t-1}} + \begin{pmatrix} \sum_{i=0}^{t-2} 2^i \cdot 2^{t-i} - 2\sum_{i=0}^{t-2} 2^i \cdot 2 \\ \sum_{i=0}^{t-2} 2^i \cdot 1 \end{pmatrix} =$$
$$= \binom{0}{2^{t-1}} + \begin{pmatrix} \sum_{i=0}^{t-2} 2^t - 2\sum_{i=0}^{t-2} 2^i \\ \sum_{i=0}^{t-2} 2^i \end{pmatrix} =$$
$$= \binom{0}{2^{t-1}} + \binom{2^t \cdot (t-1) - 2 \cdot (2^{t-1} - 1)}{2^{t-1} - 1} =$$
$$= \binom{2^t \cdot (t-1) - 2^t + 2}{2^{t-1} + 2^{t-1} - 1} = \binom{2^t \cdot (t-2) + 2}{2^t - 1}.$$

Thus, the edge functions of the model $K(2, 2^t)$, constructed in accordance with [59] contain $2^t \cdot (t-2) + 2$ conjunctions and $2^t - 1$ disjunctions, or in general $2^t \cdot (t-1) + 1$ elementary logical operations. The edge functions of the GL-model of a 1-/2-fault-tolerant n-processor system, when using the auxiliary model $K(2, n)$, constructed in accordance with [59], will be as follows:

$$f_i = c_1 f_i^{(1)} \vee c_2 f_i^{(2)} = \bar{c} x_i \vee c f_i^{(2)}, \qquad (5)$$

where $i = 1, 2, \dots, n-1$, where $f_i^{(2)}$ is the expression of the i-th function of the model $K(2, n)$, and

$$f_n = \bar{c} x_n \vee c \cdot 1 = x_n \vee c.$$

Thus, the first $n-1$ edge functions of the GL-model of a 1-/2-fault-tolerant n-processor system in addition to the expressions of the edge functions of the auxiliary model $K(2, n)$ contain 2 conjunctions and one disjunction each, and the last edge function of this model contains only one disjunction. Thus, the total complexity of the edge functions of this model will be (excluding the complexity of calculating the value of the expression c and its inversion $\bar{c}$)

$$\gamma(n) = \alpha(n) + (n-1) \cdot \binom{2}{1} + \binom{0}{1}.$$

For the above case $n = 2^t$ this value is equal to

$$\gamma(2^t) = \alpha(2^t) + (2^t - 1) \cdot \binom{2}{1} + \binom{0}{1} =$$
$$= \binom{2^t \cdot (t-2) + 2}{2^t - 1} + (2^t - 1) \cdot \binom{2}{1} + \binom{0}{1} =$$
$$= \binom{t \cdot 2^t}{2^{t+1} - 1},$$

or a total of $2^{t+1} + t \cdot 2^t - 1$ elementary operations. If we also take into account the need to perform an invert operation for the value of the expression c (once for all functions), then the total number of elementary operations will be equal to

$$\eta(2^t) = 2^{t+1} + t \cdot 2^t.$$

Note also that in some cases, expressions of the form (5) can be further simplified. For example, it can be seen that with an appropriate choice of pairs of auxiliary model functions $K(1, n)$ and $K(2, n)$, the situation of the form:

$$f_i = \bar{c} x_i \vee c(x_i \vee x_{i+1}) = \bar{c} x_i \vee c x_i \vee c x_{i+1} =$$
$$= x_i \vee c x_{i+1}.$$

Thus, instead of two additional conjunctions and one disjunction, only one conjunction is needed. It can also be shown that when $n = 2^t$ this happens for exactly 2^{t-1} edge functions (since this is the number of expressions of the functions of the model $K(2, n)$ that have the form $x_i \vee x_{i+1}$). Accordingly, the complexity of the edge functions of the GL-model of a 1-/2-fault-tolerant n-processor system using this transformation will be equal (except for the complexity of calculating the value of the expression c and its inversion $\bar{c}$).

$$\gamma'(2^t) = \alpha(2^t) + (2^t - 1 - 2^{t-1}) \cdot \binom{2}{1} +$$
$$+2^{t-1} \cdot \binom{1}{0} + \binom{0}{1} =$$
$$= \binom{2^t \cdot (t-2) + 2}{2^t - 1} + (2^{t-1} - 1) \cdot \binom{2}{1} +$$
$$+2^{t-1} \cdot \binom{1}{0} + \binom{0}{1} =$$
$$= \binom{2^t \cdot (t-1) + 2^{t-1}}{2^t + 2^{t-1} - 1} = \binom{2^{t-1} \cdot (2t-1)}{3 \cdot 2^{t-1} - 1},$$

or a total of $2^t \cdot (t+1) - 1$ elementary operations. Considering the additional inverting of the value of c, the number of elementary operations will be equal to

$$\eta'(2^t) = 2^t \cdot (t+1).$$

In some cases, other options for simplifying expressions are possible, but they are not studied in this article in general.

On the other hand, as shown above, the complexity of the edge functions of the GL-model of a 1-/2-fault-tolerant n-processor system when using the $K(2, n)$ model built in accordance with [58] as an auxiliary model (excluding the complexity of calculating the value of the expression c) is

$$\gamma''(n) = \binom{n \cdot \left\lfloor \frac{n-1}{2} \right\rfloor}{n}.$$

For the case $n = 2^t$ this value is equal to

$$\gamma''(2^t) = \binom{2^t \cdot \left\lfloor \frac{2^t - 1}{2} \right\rfloor}{2^t} = \binom{2^t \cdot \left\lfloor 2^{t-1} - \frac{1}{2} \right\rfloor}{2^t} =$$
$$= \binom{2^t \cdot (2^{t-1} - 1)}{2^t},$$

i.e., in general

$$\eta''(2^t) = 2^{2t-1}$$

elementary operations.

Table 1 contains the values of $\eta(2^t)$, $\eta'(2^t)$ and $\eta''(2^t)$ for $t = 1, 2, \ldots, 6$. As can be seen from the table, the $K(2, n)$, model built in accordance with [58] should be used for values of $t \leq 3$, which corresponds to systems (subsystems) with 8 or fewer processors. When applying the above additional simplification, the complexity of both models for $t = 3$ ($n = 8$) turns out to be the same. At the same time, as experiments show, without such a simplification, the auxiliary model built in accordance with [58] gives a lower complexity for values of $n \leq 10$ (Table 2).

Table 1. **Estimating the complexity of edge functions of GL-models of 1-/2-fault-tolerant *n*-processor systems**

t	n	$\eta(2^t)$	$\eta'(2^t)$	$\eta''(2^t)$
1	2	6	4	2
2	4	16	12	8
3	8	40	32	32
4	16	96	80	128
5	32	224	192	512
6	64	512	448	2048

Source: **compiled by the authors**

EXAMPLES

Example 1. Let us build a GL-model of a system containing 8 processors and is resistant to any 1 failure, and if the 1st and simultaneously the 2nd or 3rd processors (at least one of them) are functional, then to 2 failures. The following condition is satisfied by the Boolean expression

$$c = x_1(x_2 \vee x_3).$$

Table 2. **The complexity of edge functions of GL-models of 1-/2-fault-tolerant *n*-processor systems, calculated experimentally**

n	Complexity of the edge functions of the model	
	An auxiliary model in accordance with the [59]	An auxiliary model in accordance with the [58]
9	47	45
10	54	50
11	61	66
12	68	72
13	75	91
14	82	98
15	89	120

Source: **compiled by the authors**

When used as an auxiliary GL-model built in accordance with [58], as shown above, a model (denoted $K^{(1)}$) based on a cyclic graph with 8 edges and the following edge functions will be obtained:

$$f_1^{(1)} = x_1 \vee cx_2x_3x_4;$$
$$f_2^{(1)} = x_2 \vee cx_3x_4x_5;$$
$$f_3^{(1)} = x_3 \vee cx_4x_5x_6;$$
$$f_4^{(1)} = x_4 \vee cx_5x_6x_7;$$
$$f_5^{(1)} = x_5 \vee cx_6x_7x_8;$$
$$f_6^{(1)} = x_6 \vee cx_7x_8x_1;$$
$$f_7^{(1)} = x_7 \vee cx_8x_1x_2;$$
$$f_8^{(1)} = x_8 \vee cx_1x_2x_3.$$

It is easy to see that the expressions of the edge functions of the $K^{(1)}$ model contain a total of 8 disjunctions and 24 conjunctions (without taking into account the complexity of calculating the value of the expression c, which we consider to be calculated in advance), or a total of 32 elementary operations. This corresponds to the value of $\eta''(2^3) = 2^{2\cdot3-1} = 2^5 = 32$.

Now let us construct an auxiliary model $K(2, 8)$ according to [59]. It will contain 7 edges with the following edge functions:

$$g_1 = x_1 \vee x_2;$$
$$g_2 = x_1x_2 \vee x_3x_4;$$
$$g_3 = x_3 \vee x_4;$$
$$g_4 = x_1x_2x_3x_4 \vee x_5x_6x_7x_8;$$
$$g_5 = x_5 \vee x_6;$$

$$g_6 = x_5x_6 \vee x_7x_8;$$
$$g_7 = x_7 \vee x_8.$$

Also, as was shown in [60] we supplement this model with an edge with an additional trivial edge function

$$g_8 = 1.$$

The model (denoted as $K^{(2)}$) of the system under consideration, thus, according to [59] will be based on a cyclic graph with 8 edges and the following edge functions:

$$f_1^{(2)} = \bar{c}x_1 \vee cg_1 = \bar{c}x_1 \vee c(x_1 \vee x_2);$$
$$f_2^{(2)} = \bar{c}x_2 \vee cg_2 = \bar{c}x_2 \vee c(x_1x_2 \vee x_3x_4);$$
$$f_3^{(2)} = \bar{c}x_3 \vee cg_3 = \bar{c}x_3 \vee c(x_3 \vee x_4);$$
$$f_4^{(2)} = \bar{c}x_4 \vee cg_4 =$$
$$= \bar{c}x_4 \vee c(x_1x_2x_3x_4 \vee x_5x_6x_7x_8);$$
$$f_5^{(2)} = \bar{c}x_5 \vee cg_5 = \bar{c}x_5 \vee c(x_5 \vee x_6);$$
$$f_6^{(2)} = \bar{c}x_6 \vee cg_6 = \bar{c}x_6 \vee c(x_5x_6 \vee x_7x_8);$$
$$f_7^{(2)} = \bar{c}x_7 \vee cg_7 = \bar{c}x_7 \vee c(x_7 \vee x_8);$$
$$f_8^{(2)} = \bar{c}x_8 \vee cg_8 = \bar{c}x_8 \vee c = x_1 \vee c.$$

As you can see, the edge functions of the $K^{(2)}$ model, contain a total of 15 disjunctions and 24 conjunctions. In addition, you need to perform one operation to invert the value of the expression c. Thus, a total of 40 elementary logical operations must be performed, which corresponds to the above expression $\eta(2^3) = 2^{3+1} + 3 \cdot 2^3 = 16 + 3 \cdot 8 = 40$.

If we further simplify the expressions of the edge functions of the model (thus obtaining the $K^{(3)}$ model), they will take the following form:

$$f_1^{(3)} = x_1 \vee cx_2;$$
$$f_2^{(3)} = \bar{c}x_2 \vee c(x_1x_2 \vee x_3x_4);$$
$$f_3^{(3)} = x_1 \vee cx_4;$$
$$f_4^{(2)} = \bar{c}x_5 \vee c(x_1x_2x_3x_4 \vee x_5x_6x_7x_8);$$
$$f_5^{(3)} = x_5 \vee cx_6;$$
$$f_6^{(3)} = \bar{c}x_6 \vee c(x_5x_6 \vee x_7x_8);$$
$$f_7^{(3)} = x_7 \vee cx_8;$$
$$f_8^{(3)} = x_1 \vee c.$$

The expressions of the edge functions of the $K^{(3)}$ model contain 11 disjunctions, 20 conjunctions, and 1 inversion, which gives a total of 32 elementary operations. This corresponds to the meaning of the expression $\eta'(2^3) = 2^3 \cdot (3 + 1) = 8 \cdot 4 = 32$.

All of the above models are based on cyclic graphs and differ only in the expressions of the edge functions. As can be seen, the overall complexity of the edge functions in the $K^{(1)}$ and $K^{(3)}$ models is the same, and the $K^{(2)}$ model is more complex than the previous ones, which corresponds to the values given in Table 1.Experiments show that all 3 models $K^{(1)}$, $K^{(2)}$ and $K^{(3)}$ show the system's operable state on all vectors with no more than 1 zero, as well as on 20 vectors with 2 zeros, shown in Table 3. As you can see, vectors with numbers 1...10 correspond to all possible situations when there are 2 failures in the system, and the 1st, 2nd and 3rd processors are in good condition; vectors with numbers 11...15 correspond to all possible situations when there are 2 failures in the system, with the 1st and 2nd processors being serviceable and the 3rd processor being faulty; and vectors 16...20 correspond to all possible situations when there are 2 failures in the system, with the 1st and 3rd processors being serviceable and the 2nd processor being faulty. Thus, the model really reflects the system's resilience to two failures only and only in situations where the condition «the 1st and simultaneously the 2nd or 3rd processors are serviceable' is met».

It is also worth noting that although the models show the same behavior of the system in the failure flow, some vectors lose a different number of edges. For example, on vectors numbered 4, 12, and 18 in Table 3, models $K^{(2)}$ and $K^{(3)}$ lose one edge each, while model $K^{(1)}$ does not lose any edges.

Example 2. Let us build a GL-model of a system containing 10 processors that is resilient to any 1 failure, and under some condition (which corresponds to the Boolean expression c), to 2 failures. The specific type of Boolean expression c is not important in this example.

Let us build a GL-model of such a system, using the $K(2, 10)$, model built in accordance with [58] as an auxiliary model. The resulting $K^{(1)}$ model will be based on a cyclic graph with 10 edges, which will be represented by the following edge functions:

$$f_1^{(1)} = x_1 \vee cx_2x_3x_4x_5;$$
$$f_2^{(1)} = x_2 \vee cx_3x_4x_5x_6;$$
$$f_3^{(1)} = x_3 \vee cx_4x_5x_6x_7;$$
$$f_4^{(1)} = x_4 \vee cx_5x_6x_7x_8;$$
$$f_5^{(1)} = x_5 \vee cx_6x_7x_8x_9;$$
$$f_6^{(1)} = x_6 \vee cx_7x_8x_9x_{10};$$
$$f_7^{(1)} = x_7 \vee cx_8x_9x_{10}x_1;$$
$$f_8^{(1)} = x_8 \vee cx_9x_{10}x_1x_2;$$
$$f_9^{(1)} = x_9 \vee cx_{10}x_1x_2x_3;$$
$$f_{10}^{(1)} = x_{10} \vee cx_1x_2x_3x_4.$$

Table 3. **The number of edges that the model loses on blocked vectors with 2 zeros**

No.	System state vector	Number of lost edges of the model		
		$K^{(1)}$	$K^{(2)}$	$K^{(3)}$
1	**111**11100	1	1	1
2	**111**11010	1	1	1
3	**111**10110	1	1	1
4	**111**01110	0	1	1
5	**111**11001	1	1	1
6	**111**10101	1	1	1
7	**111**01101	1	1	1
8	**111**10011	1	1	1
9	**111**01011	1	1	1
10	**111**00111	1	1	1
11	**11**011110	1	1	1
12	**11**011101	0	1	1
13	**11**011011	1	1	1
14	**11**010111	1	1	1
15	**11**001111	1	1	1
16	**10**111110	1	1	1
17	**10**111101	1	1	1
18	**10**111011	0	1	1
19	**10**110111	1	1	1
20	**10**101111	1	1	1

Source: **compiled by the authors**

The edge functions of the $K^{(1)}$ model contain 10 disjunctions and 40 conjunctions, i.e., a total of 50 elementary operations, which corresponds to the value in Table 2.

The $K^{(2)}$ model of the same system, which uses the $K(2, 10)$, model as an auxiliary model, constructed in accordance with [59] will also be based on a cyclic graph with 10 edges and the following edge functions:

$$f_1^{(2)} = \bar{c}x_1 \vee c(x_1 \vee x_2);$$
$$f_2^{(2)} = \bar{c}x_2 \vee c(x_1x_2 \vee x_3);$$
$$f_3^{(2)} = \bar{c}x_3 \vee c(x_1x_2x_3 \vee x_4x_5);$$
$$f_4^{(2)} = \bar{c}x_4 \vee c(x_4 \vee x_5);$$
$$f_5^{(2)} = \bar{c}x_5 \vee c(x_1x_2x_3x_4x_5 \vee x_6x_7x_8x_9x_{10});$$
$$f_6^{(2)} = \bar{c}x_6 \vee c(x_6 \vee x_7);$$
$$f_7^{(2)} = \bar{c}x_7 \vee c(x_6x_7 \vee x_8);$$
$$f_8^{(2)} = \bar{c}x_8 \vee c(x_6x_7x_8 \vee x_9x_{10});$$
$$f_9^{(2)} = \bar{c}x_9 \vee c(x_9 \vee x_{10});$$
$$f_{10}^{(2)} = x_{10} \vee c.$$

The edge functions of the model contain 19 disjunctions, 34 conjunctions, and 1 inversion, which total 54 elementary logical operations. This also corresponds to the value given in Table 2.

After simplifying the expressions of the edge functions of the $K^{(2)}$ model, we obtain the $K^{(3)}$ model with the following edge functions:

$$f_1^{(3)} = x_1 \vee cx_2;$$
$$f_2^{(3)} = \bar{c}x_2 \vee c(x_1x_2 \vee x_3);$$
$$f_3^{(3)} = \bar{c}x_3 \vee c(x_1x_2x_3 \vee x_4x_5);$$
$$f_4^{(3)} = x_4 \vee cx_5;$$
$$f_5^{(3)} = \bar{c}x_5 \vee c(x_1x_2x_3x_4x_5 \vee x_6x_7x_8x_9x_{10});$$
$$f_6^{(3)} = x_6 \vee cx_7;$$
$$f_7^{(3)} = \bar{c}x_7 \vee c(x_6x_7 \vee x_8);$$
$$f_8^{(3)} = \bar{c}x_8 \vee c(x_6x_7x_8 \vee x_9x_{10});$$
$$f_9^{(3)} = x_9 \vee cx_{10};$$
$$f_{10}^{(3)} = x_{10} \vee c.$$

As can be seen, the edge functions of the $K^{(3)}$ model contain 15 disjunctions, 30 conjunctions and 1 inversion, i.e., 46 elementary logical operations in total. Given that all models are based on the same cyclic graphs with 10 edges, we can conclude that the complexity of model $K^{(3)}$ is lower than that of model $K^{(1)}$, but the complexity of model $K^{(2)}$ is higher than that of model $K^{(1)}$. Note also that, as can be seen in Table 2, the complexity of the $K^{(2)}$ model would also be lower than the complexity of the $K^{(1)}$ model, if the system contained 11 or more processors.

It is also worth noting that this study does not analyze, in general, additional possibilities for simplifying the expressions of the edge functions of the resulting system, which can be achieved, in particular, by rearranging the edges of the auxiliary models. Thus, for example, as a result of the appropriate rearrangement of the edges of the auxiliary model $K(2, 10)$, the model $K^{(4)}$ could contain the following edge functions:

$$f_1^{(4)} = \bar{c}x_1 \vee c(x_1 \vee x_2);$$
$$f_2^{(4)} = \bar{c}x_2 \vee c(x_1x_2x_3 \vee x_4x_5);$$
$$f_3^{(4)} = \bar{c}x_3 \vee c(x_1x_2 \vee x_3);$$
$$f_4^{(4)} = \bar{c}x_4 \vee c(x_4 \vee x_5);$$
$$f_5^{(4)} = \bar{c}x_5 \vee c(x_1x_2x_3x_4x_5 \vee x_6x_7x_8x_9x_{10});$$
$$f_6^{(4)} = \bar{c}x_6 \vee c(x_6 \vee x_7);$$
$$f_7^{(4)} = \bar{c}x_7 \vee c(x_6x_7x_8 \vee x_9x_{10});$$
$$f_8^{(4)} = \bar{c}x_8 \vee c(x_6x_7 \vee x_8);$$
$$f_9^{(4)} = \bar{c}x_9 \vee c(x_9 \vee x_{10});$$
$$f_{10}^{(4)} = x_{10} \vee c.$$

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

It is easy to see that the total complexity of the expressions of the edge functions of the $K^{(4)}$ model does not differ from the total complexity of the expressions of the edge functions of the $K^{(2)}$ model. However, by simplifying the expressions of the edge functions of the $K^{(4)}$ model, we can obtain the $K^{(5)}$ model with the following edge functions:

$$f_1^{(5)} = x_1 \vee cx_2;$$
$$f_2^{(5)} = \bar{c}x_2 \vee c(x_1x_2x_3 \vee x_4x_5);$$
$$f_3^{(5)} = x_3 \vee cx_1x_2;$$
$$f_4^{(5)} = x_4 \vee cx_5;$$
$$f_5^{(5)} = \bar{c}x_5 \vee c(x_1x_2x_3x_4x_5 \vee x_6x_7x_8x_9x_{10});$$
$$f_6^{(5)} = x_6 \vee cx_7;$$
$$f_7^{(5)} = \bar{c}x_7 \vee c(x_6x_7x_8 \vee x_9x_{10});$$
$$f_8^{(5)} = x_8 \vee cx_6x_7;$$
$$f_9^{(5)} = x_9 \vee cx_{10};$$
$$f_{10}^{(5)} = x_{10} \vee c.$$

The edge functions of the $K^{(5)}$ model contain 13 disjunctions, 28 conjunctions, and 1 inversion, i.e., a total of 42 elementary operations. Thus, the complexity of the $K^{(5)}$ model is even lower than that of the $K^{(3)}$ model.

Example 3. Let us build a GL-model of a system containing 7 processors and resilient to any 1 failure, and under some condition (which corresponds to the Boolean expression *c*), to 2 failures.

GL-model $K^{(1)}$ of this system, when used as an auxiliary model $K(2, 7)$, constructed in accordance with [58], will be based on a cyclic graph with 10 edges with the following edge functions:

$$f_1^{(1)} = x_1 \vee cx_2x_3x_4;$$
$$f_2^{(1)} = x_2 \vee cx_3x_4x_5;$$
$$f_3^{(1)} = x_3 \vee cx_4x_5x_6;$$
$$f_4^{(1)} = x_4 \vee cx_5x_6x_7;$$
$$f_5^{(1)} = x_5 \vee cx_6x_7x_1;$$
$$f_6^{(1)} = x_6 \vee cx_7x_1x_2;$$
$$f_7^{(1)} = x_7 \vee cx_1x_2x_3.$$

If the $K(2, 7)$, model constructed in accordance with [59], is used as an auxiliary model, then a $K^{(2)}$ model based on a cyclic graph with 7 edges and the following edge functions will be obtained:

$$f_1^{(2)} = \bar{c}x_1 \vee c(x_1 \vee x_2);$$
$$f_2^{(2)} = \bar{c}x_2 \vee c(x_1x_2 \vee x_3x_4);$$
$$f_3^{(2)} = \bar{c}x_3 \vee c(x_3 \vee x_4);$$
$$f_4^{(2)} = \bar{c}x_4 \vee c(x_1x_2x_3x_4 \vee x_5x_6x_7);$$
$$f_5^{(2)} = \bar{c}x_5 \vee c(x_5 \vee x_6);$$
$$f_6^{(2)} = \bar{c}x_6 \vee c(x_5x_6 \vee x_7);$$
$$f_7^{(2)} = x_7 \vee c.$$

The model $K^{(3)}$ obtained by simplifying the expressions of the edge functions of the model $K^{(2)}$ will have the following edge functions:

$$f_1^{(3)} = x_1 \vee cx_2;$$
$$f_2^{(3)} = \bar{c}x_2 \vee c(x_1x_2 \vee x_3x_4);$$
$$f_3^{(3)} = x_3 \vee cx_4;$$
$$f_4^{(3)} = \bar{c}x_4 \vee c(x_1x_2x_3x_4 \vee x_5x_6x_7);$$
$$f_5^{(3)} = x_5 \vee cx_6;$$
$$f_6^{(3)} = \bar{c}x_6 \vee c(x_5x_6 \vee x_7);$$
$$f_7^{(3)} = x_7 \vee c.$$

If we change the order of the edge functions when constructing the $K^{(2)}$, model, we can obtain the $K^{(4)}$ model with the following edge functions:

$$f_1^{(4)} = \bar{c}x_1 \vee c(x_1 \vee x_2);$$
$$f_2^{(4)} = \bar{c}x_2 \vee c(x_1x_2 \vee x_3x_4);$$
$$f_3^{(4)} = \bar{c}x_3 \vee c(x_3 \vee x_4);$$
$$f_4^{(4)} = \bar{c}x_4 \vee c(x_1x_2x_3x_4 \vee x_5x_6x_7);$$
$$f_5^{(4)} = \bar{c}x_5 \vee c(x_5 \vee x_6);$$
$$f_6^{(4)} = x_6 \vee c;$$
$$f_7^{(4)} = \bar{c}x_7 \vee c(x_5x_6 \vee x_7).$$

Further, as a result of simplifying the expressions of the edge functions of model $K^{(4)}$ we obtain model $K^{(5)}$, which contains the following edge functions:

$$f_1^{(5)} = x_1 \vee cx_2;$$
$$f_2^{(5)} = \bar{c}x_2 \vee c(x_1x_2 \vee x_3x_4);$$
$$f_3^{(5)} = x_3 \vee cx_4;$$
$$f_4^{(5)} = \bar{c}x_4 \vee c(x_1x_2x_3x_4 \vee x_5x_6x_7);$$
$$f_5^{(5)} = x_5 \vee cx_6;$$
$$f_6^{(5)} = x_6 \vee c;$$
$$f_7^{(5)} = x_7 \vee cx_5x_6.$$

All of the above models are based on the same cyclic graphs with 7 edges and differ only in the expressions of the edge functions. The complexity of the expressions of these edge functions is given in Table 4. As can be seen, the $K^{(2)}$ model is, as expected, more complex than the $K^{(1)}$ model, and the $K^{(3)}$ – model is at least as complex as it is. However, the model $K^{(5)}$ obtained as a result of additional simplifications is still somewhat simpler than model $K^{(1)}$.

Table 4. **Number of logical operations in the expressions of edge functions of GL-models from Example 3**

Logical operations	Model				
	$K^{(1)}$	$K^{(2)}$	$K^{(3)}$	$K^{(4)}$	$K^{(5)}$
Disjunctions	7	13	10	13	9
Conjunctions	21	20	17	20	16
Inversions	0	1	1	1	1
Overall	28	34	28	34	26

Source: **compiled by the authors**

CONCLUSIONS

This paper considers the application of the method described in [60] to build GL-models of FTMS that are resistant to the failure of any one of their processors, and, under certain conditions, to the failure of two of their processors. In this case, both models built in accordance with [59], and models built in accordance with [58] can be used as auxiliary models. The resulting GL-models are based on identical cyclic graphs with n edges, where n – is the number of processors in the system.

The expressions for estimating the complexity of expressions of edge functions of models constructed by each of the methods are obtained, in particular for cases when the number of processors is a power of two.

Experiments were conducted involving the explicit construction of GL models using the methods discussed in the paper, followed by the calculation of the number of operations in the expressions of their edge functions. The results confirm the analytically derived complexity estimates.

It is shown that the use of models constructed in accordance with [58] is appropriate if the number of processors is no more than 10, and when using an additional simplification of the expressions of edge functions – if it does not exceed 8. However, in some cases, other simplifications are possible that allow achieving lower model complexity when using models built in accordance with [59], as auxiliary models, which can be investigated in more detail in the future.

In addition, GL-models built in different ways differ in the ratio between the numbers of logical operations of different types (conjunctions, disjunctions) used in the expressions of their edge functions, which may also be important sometimes. When used as auxiliary models constructed in accordance with [58] all expressions of the model's edge functions have the same structure, unlike in the case of using models constructed in accordance with [59]. This can be important, for example, when implementing parallel calculation of such values.

The experiments show that all the constructed GL-models adequately reflect the behavior of the system in the fault flow. However, the models built with the use of auxiliary models constructed in accordance with [58] on some vectors may lose a smaller number of edges compared to the case when the auxiliary models are constructed in accordance with [59]. This property can be important, in particular, if additional modifications of the built GL-model are planned.

It is also worth noting that the models constructed by the methods described above can be used not only independently, but also as components/auxiliary models for building models of more complex systems, for example [61, 62].

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 smportant

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

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 – 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. 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.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.

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

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. 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. 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.

21. 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.

22. 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.

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-n 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. 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.

41.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.

42. 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.

43.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.

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. 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.

48. 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.

49. 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.

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.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.

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. 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; 7 (2): 175–188. DOI: https://doi.org/10.15276/aait.07.2024.13.

61. Romankevich A. M., Morozov K. V. & Romankevich V. A. "Graph-Logic models of hierarchical fault-tolerant multiprocessor systems". *IJCSNS International Journal of Computer Science and Network Security*. 2019; 19 (7): 151–156.

62. 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.

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

Received 24.12.2024
Received after revision 10.03.2025
Accepted 17.03.2025

DOI: https://doi.org/10.15276/aait.08.2025.9
УДК 004.05

GL-моделі для аналізу багатопроцесорних систем, стійких до однієї та двох відмов

Романкевич Віталій Олексійович [1)]
ORCID: https://orcid.org/0000-0003-4696-5935; zavkaf@scs.kpi.ua. Scopus Author ID: 57193263058
Морозов Костянтин В'ячеславович [1)]
ORCID: https://orcid.org/0000-0003-0978-6292; mcng@ukr.net. Scopus Author ID: 57222509251
Романкевич Олексій Михайлович [1)]
ORCID: https://orcid.org/0000-0001-5634-8469; romankev@scs.kpi.ua. Scopus Author ID: 6602114176
Галицький Даніл Володимирович [1)]
ORCID: https://orcid.org/0009-0004-4421-3443; zipper135401@gmail.com. Scopus Author ID: 58553487600
Журба Антон Володимирович [1)]
ORCID: https://orcid.org/0009-0007-3375-2590; zhurba.anton@outlook.com
[1)] Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна

АНОТАЦІЯ

Роботу присвячено проблемі побудови GL-моделей поведінки в потоці відмов небазових відмовостійких багатопроцесорних систем. Розглядаються системи, які є стікими до будь-якої однієї відмови, а за певної умови – до будь-яких двох відмов своїх процесорів. При цьому передбачається, що такій умові відповідає деякий булевий вираз, що залежить від станів процесорів системи. Для побудови моделей таких систем застосовується запропонований раніше спосіб, який базується на комбінуванні виразів реберних функцій допоміжних базових GL-моделей, що відповідають 1- та 2-відмовостійким системам. Розглядається два альтернативних способи побудови допоміжних моделей $K(2, n)$. Показано, що отримані GL-моделі базуються на однакових циклічних графах, і відрізняються лише виразами своїх реберних функцій. Виконано аналіз складності виразів реберних функцій отриманих моделей, для випадків використання допоміжних GL-моделей кожного типу. Отримано формули для оцінки складності цих виразів (кількості елементарних логічних операцій) в залежності від кількості процесорів в системі, зокрема, для випадків, коли ця кількість є ступенем двійки. Також, для випадку використання однієї з допоміжних моделей, виявлено можливість додаткового спрощення виразів деяких із реберних функцій. Визначено, в яких випадках, а саме, для якої кількості процесорів системи, доцільно використовувати допоміжні моделі того чи іншого типу. Проведені експерименти підтверджують, що побудовані розглянутим способом GL-моделі адекватно відображають поведінку системи. Наведено приклади, що демонструють застосування розглянутого способу побудови небазових GL-моделей та підтверджують коректність оцінки складності виразів їх реберних функцій при використанні допоміжних моделей різних типів. Показано також, що в деяких випадках існує можливість додаткового спрощення виразів реберних функцій GL-моделей, що будуються, зокрема за рахунок зміни порядку ребер у допоміжних моделях. Проте, в рамках даної статті цю можливість не досліджено в загальному випадку.

Ключові слова: відмовостійкість; багатопроцесорні системи; оцінка надійності; GL-моделі; небазові системи; циклічний граф; реберні функції

ABOUT THE AUTHORS

Vitaliy A. Romankevich - Doctor of Engineering Sciences, Professor, Head of System Programming and Specialized Computer System Department, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Peremogy Ave. Kyiv, 03056, Ukraine
ORCID: https://orcid.org/0000-0003-4696-5935; zavkaf@scs.kpi.ua. Scopus Author ID: 57193263058
***Research field*:** Fault-tolerant multiprocessor systems reliability estimation, GL-models, Self-diagnosable systems, Diagnosis of multiprocessor systems, Discrete mathematics

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

Kostiantyn V. Morozov - PhD (Eng), Associate Professor, System Programming and Specialized Computer System Department, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Peremogy Ave. Kyiv, 03056, Ukraine
ORCID: https://orcid.org/0000-0003-0978-6292; mcng@ukr.net. Scopus Author ID: 57222509251
***Research field*:** GL-models, Fault-tolerant multiprocessor systems reliability estimation, Self-diagnosable multiprocessor systems.

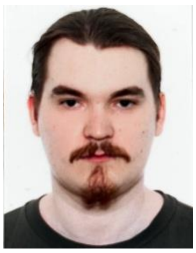
Морозов Костянтин В'ячеславович - кандидат технічних наук, доцент кафедри Системного програмування та спеціалізованих комп'ютерних систем. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна

Alexei M. Romankevich - Doctor of Engineering Sciences, Professor, System Programming and Specialized Computer System Department, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Peremogy Ave. Kyiv, 03056, Ukraine
ORCID: https://orcid.org/0000-0001-5634-8469; romankev@scs.kpi.ua. Scopus Author ID: 6602114176
***Research field*:** Fault-tolerant multiprocessor systems reliability estimation, GL-models, Self-diagnosable systems, Diagnosis of multiprocessor systems, Multi-valued logic.

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

Daniil V. Halytsky - postgraduate student, System Programming and Specialized Computer System Department, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Peremogy Ave. Kyiv, 03056, Ukraine
ORCID: https://orcid.org/0009-0004-4421-3443; zipper135401@gmail.com. Scopus Author ID: 58553487600
***Research field*:** GL-models building and processing optimization, Parallel computing and multi-core system optimization, Algorithm design for distributed and high-performance computing, Graph algorithms and their applications in complex systems, Computational efficiency in large-scale data processing.

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



Anton V. Zhurba - postgraduate student, System Programming and Specialized Computer System Department, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Peremogy Ave. Kyiv, 03056, Ukraine
ORCID: https://orcid.org/0009-0007-3375-2590; zhurba.anton@outlook.com
***Research field*:** Testing and verification of parallel systems, Parallel computing and optimization of algorithms, Mathematical analysis of performance, Modeling and simulation of computing processes.

Журба Антон Володимирович - аспірант, кафедра Системного програмування та спеціалізованих комп'ютерних систем, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)