DOI: https://doi.org/10.15276/aait.08.2025.15
UDC 004.05

Generalization of the method for constructing GL-models of complex fault-tolerant multiprocessor systems with additional failure conditions

Vitaliy A. Romankevich[1)]
ORCID: https://orcid.org/0000-0003-4696-5935; zavkaf@scs.kpi.ua. Scopus Author ID: 57193263058
Ihor A. Yermolenko[1)]
ORCID: https://orcid.org/0009-0008-5298-4888; yermolenkomail@gmail.com
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
Oleksandr O. Melnyk[1)]
ORCID: https://orcid.org/0009-0003-7527-8146; inboxnumberzero@zoho.com
[1)] National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Peremogy Ave. Kyiv, 03056, Ukraine

ABSTRACT

The article is devoted to methods for constructing GL-models of fault-tolerant multiprocessor systems. GL-models can be used as behavioral models of such systems under failure flows to evaluate their reliability metrics through statistical experiments. The study considers two types of systems: consecutive two-dimensional systems and mixed-type systems. A consecutive two-dimensional system is defined as one in which components are arranged in the form of a rectangular matrix, and the system fails when a rectangular block of a certain size appears, consisting entirely of failed components. A mixed-type system fails if at least one of the following conditions is met: a specified number of arbitrary components have failed; a specified number of consecutive components have failed; or a rectangular block of a certain size, consisting entirely of failed components, appears within the component matrix. Currently, there are no formalized methods for constructing GL-models for the aforementioned types of systems. The objective of this work is to develop a universal method for constructing GL-models for both consecutive two-dimensional systems and mixed-type systems. It is shown that, to construct a GL-model for such a system, it is sufficient to determine the maximum number of failed components under which the system remains operational. Based on this threshold, a basic system model is constructed without considering additional failure conditions. Then, all combinations of component failures that lead to system failure are identified. The basic model is subsequently weakened at the vectors corresponding to these critical failure combinations. This paper presents, for the first time, an algorithm for constructing GL-models for consecutive two-dimensional systems and mixed-type systems. In addition, it introduces methods for calculating the maximum allowable number of component failures under which the system remains functional, as well as estimating the total number of failure combinations that result in system failure. Experimental results confirm that the proposed models adequately represent the real behavior of such systems under failure flows. Examples are provided to illustrate the GL-model construction process for both of the aforementioned system types.

Keywords: GL-models; non-basic fault-tolerant multiprocessor systems; consecutive two-dimensional system; mixed-type systems; reliability assessment

For citation: **Romankevich V. A., Yermolenko I. A., Morozov K. V., Romankevich A.M., Melnyk O. O. "Generalization of the method for constructing GL-models of complex fault-tolerant multiprocessor systems with additional failure conditions".** ***Applied Aspects of Information Technology*****. 2025; Vol. 8 No.2: 216–230. DOI: https://doi.org/10.15276/aait.08.2025.15**

INTRODUCTION

Human involvement in the control of complex systems has significantly reduced by modern automated control systems (CS) [1], [2]. The reduced influence of the human factor has improved the stability of such systems, and the operator has been freed from monotonous routine tasks. Moreover, the performance of such systems has significantly increased, allowing them to solve problems with high computational complexity. Such CS are based on microprocessor systems capable of receiving signals from control devices, sensors, or monitoring systems, processing them, and generating appropriate control signals depending on the received information.

Failures of control systems in fields such as aviation, space industry, energy, and critical infrastructure objects can cause significant material damage and financial losses. Equally critical are failures in CS for vehicles and aircraft using autopilots. Since these systems operate autonomously and make decisions based solely on information from the environment, the failure of a component can lead to the loss of the vehicle or even fatal consequences.

Therefore, it is essential for the above-mentioned systems to be capable of continuously performing their specified functions for a certain period under defined conditions. In other words, a certain level of reliability must be ensured for such systems.

However, failures can occur even in the most reliable systems. Thus, fault tolerance plays a vital role – namely, the system's ability to maintain operability or quickly recover in the event of partial failures.

Given the critical importance of maintaining the required level of reliability in such systems, fault-tolerant multiprocessor systems (FTMS) are used to build their CS. Such systems consist of many processors and can continue operating even when some of them fail. In the design of FTMS, significant attention is paid to reliability and safety calculations.

LITERATURE REVIEW AND PROBLEM STATEMENT

A fault-tolerant multiprocessor systems is referred to as basic if it continues to operate when the number of failures does not exceed m, or, in other words, when the number of functioning processors is at least $n - m$, where n is the total number of processors in the system. Non-basic systems, in contrast, can demonstrate different behavior under the same number of failures, depending on their combination: the system may fail for one failure combination and remain operable for another combination of failures of the same multiplicity. In several studies, basic systems are referred to as k-out-of-n: F or k-out-of-n:G systems – such systems fail when k components fail, or, conversely, remain operational when k components continue to function. In this paper, the notation k-out-of-n will refer to k-out-of-n:F systems. It is important to note that many real-world systems are not basic.

There exists a wide range of analytical methods for evaluating the reliability of both basic and non-basic systems. Depending on the specific configuration – and, accordingly, the conditions under which the system fails – the following types of systems can be distinguished: k-out-of-n systems [3], [4], [5]; consecutive-k-out-of-n [6], [7], [8]; consecutive-k-within-m-out-of-n [9], [10], [11]; consecutive-k-out-of-r-from-n [12], [13], [14]; m-consecutive-k-out-of-n [15], [16], [17], [18]; (n, f, k) systems [19], [20], [21]; $\langle n, f, k\rangle$ systems [20], [21]; m-consecutive-k, l-out-of-n [22], [23], [24]; k_c-out-of-n [17], [18]; consecutive-(r, s)-out-of-(m, n) [25], [26], [27]; consecutive-k_r-out-of-n_r [28]; and others. However, the main drawback of analytical approaches is the need to develop new methods each time the failure condition changes, or new conditions are introduced.

In addition to analytical methods for evaluating the reliability of FTMS, there are also methods based on statistical experiments with models of system behavior in the failure flow. Despite the drawback of such methods – namely, the accuracy of the reliability parameter estimate depends on the number of experiments – this approach is considered universal. GL-models [29], [30] can be used as models of FTMS behavior in the failure flow. A GL-model is a undirected graph in which each edge is assigned a predefined Boolean function. The Boolean function takes arguments x_i (elements of the system state vector), each of which represents the state of a corresponding processor in the system. The value of x_i is 1 if the processor is operational, and 0 if the processor has failed. If the Boolean function assigned to an edge evaluates to 0, the edge is removed from the graph. The loss of graph connectivity corresponds to the failure of the entire system.

Like FTMS, GL-models can be classified as basic or non-basic. A basic GL-model corresponds to the behavior of a system in the failure flow that consists of n components and remains operational if no more than m failures occur ($n > m$). Accordingly, the graph of a basic GL-model loses connectivity when the system state vector contains $m + 1$ or more zeros.

A non-basic GL-model can be obtained by modifying a basic model. As a result, the model's behavior changes compared to the basic model on certain system state vectors, and the model accurately reflects the system's behavior in the failure flow. A change in model behavior on a specific state vector is referred to as the blocking of that vector. According to [31], we consider that a model $K(m, n)$ can be modified either by weakening – so that its graph loses connectivity on some vectors containing m or fewer zeros – or by strengthening, meaning the graph retains connectivity on some vectors containing more than m zeros. A system can be modified in several ways: by changing the expressions of the edge functions, by altering the graph structure, or by combining both approaches.

When additional system failure conditions are introduced – such as the failure of a specific group of processors or the failure of any sequence of k consecutively connected processors – it is relatively easy to represent these conditions in a GL-model, whereas analytical methods may require significant recalculations. It is also fairly straightforward to construct a GL-model for a system composed of subsystems, each of which fails under different conditions, while developing an analytical method for such a system can prove to be a rather complex task.

Despite the universality of GL-models and the ubiquity of non-basic k-out-of-n systems, there is no universal method for constructing GL-models for these types of systems. Thus, the aim of this work is to develop methods for constructing GL-models for certain types of non-basic k-out-of-n systems.

PURPOSE AND OBJECTIVES OF THE RESEARCH

In [32], methods for constructing GL-models were proposed for consecutive-k-out-of-n and (n, f, k) systems. However, systems of the consecutive-(r, s)-out-of-(m, n) type [25], [26], [27] were not considered in that study, leaving open the question of whether GL-models can be constructed for this type of system. The consecutive-(r, s)-out-of-(m, n) system reflects a two-dimensional arrangement of processors, as, for example, in multiprocessor matrix structures. This type of system allows for more complex failure conditions compared to classical configurations such as consecutive-k-out-of-n or k-out-of-n.

Mixed systems that combine the properties of several types of systems deserve special attention. An example of such a system is a combined system of the k-out-of-n, consecutive-k-out-of-n, and consecutive-(r, s)-out-of-(m, n) types [33]. Since the failure of such a system can be caused by the fulfillment of one of three different conditions, the calculation of its reliability indicators is more complicated compared to systems that have only one type of failure condition. Real-world systems [33] may be of a mixed type, which makes the analysis of such systems relevant from both theoretical and practical perspectives.

Therefore, the purpose of the study is to create universal methods for constructing GL-models for such type of systems as consecutive-(r, s)-out-of-(m, n) systems and for mixed of the k-out-of-n, consecutive-k-out-of-n, and consecutive-(r, s)-out-of-(m, n) systems. To this end, formulae for calculating the model parameters will be derived, algorithms for constructing GL-models for the aforementioned types of systems will be developed, and the adequacy of these models in representing system behavior under failure flow will be experimentally assessed.

MLE-MODELS

A GL-model of a basic system consisting of N processors that remain operational under no more than m failures will be denoted as $K(m, N)$. The method for constructing GL-models for consecutive-(r, s)-out-of-(m, n) systems are based on the use of MLE-models (minimum lost edges) [34]. In [32], a method for constructing a GL-model for a consecutive-k-out-of-n system is described, which also uses on MLE-models as its basis. Therefore, the method for constructing a GL-model for a mixed system of the k-out-of-n, consecutive-k-out-of-n, and consecutive-(r, s)-out-of-(m, n) types will also be based on MLE-models.

One of the features of MLE-models is that the model's graph loses exactly one edge when m failures occur, and exactly two edges when $m + 1$ failures occur. Since the graph of a basic model is cyclic, the graph loses connectivity when $m + 1$ failure occur. In turn, the graph does not lose any edges if the number of failures is less than m. The number of lost edges can be calculated using the formula:

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

The total number of edges in the graph – and, accordingly, the number of edge functions – can be calculated using formula [35]:

$$\rho(m,n) = n - m + 1. \qquad (2)$$

COMMON APPROACH

Let's consider a common approach for constructing GL-models of consecutive-(r, s)-out-of-(m, n) systems and mixed systems. We begin by determining the maximum number of failures under which the system remains operational. It is easy to observe that for a consecutive-(r, s)-out-of-(m, n) system, the maximum number of failures under which the system remains operational can be achieved when the system is conditionally divided into blocks of size $r \times s$, assuming that only one functioning element remains in each such block. For a mixed system, it is sufficient to calculate the maximum allowable number of failures for each condition separately (M_1, M_2, M_3) and determine $M_{\min}$:

$$M_{\min} = \min(M_1, M_2, M_3)\,. \tag{3}$$

Next, we construct the MLE-model $K(M_{\min}, N)$.

The resulting MLE-model is modified to take into account additional system failure conditions: failure of a block of elements of size $r \times s$, failure of k consecutive elements, etc. To do this, we identify the vectors on which the graph must lose connectivity and block them by weakening the model. This can be done by modifying the expressions of the model's edge functions, altering the structure of its graph, or by combining both approaches. The resulting GL-model will correspond the behavior of the given system in the failure flow.

CONSECUTIVE-(R, S)-OUT-OF-(M, N) SYSTEMS

Consecutive-(r, s)-out-of-(m, n) systems are a generalization of k-out-of-n systems. In fact, such a system can be viewed as a two-dimensional modification of a consecutive-k-out-of-n:F system, i.e., a system consisting of n elements that fails when k consecutive elements fail. In the case of a consecutive-(r, s)-out-of-(m, n):F system, the system consists of m rows and n columns. The failure condition for such a system is the failure of a block of elements of size $r \times s$ (where $1 \le r \le m$, $1 \le s \le n$). A consecutive-(r, s)-out-of-(m, n):G system, in contrast, continues to function only if there exists at least one block of size $r \times s$ in which all elements remain operational. In this paper, we consider consecutive-(r, s)-out-of-(m, n):F systems, and we will refer to them simply as consecutive-(r, s)-out-of-(m, n).

To construct an auxiliary MLE-model $K(M, N)$, it is necessary to determine the maximum allowable number of failures M under which the system remains operational. The value of the maximum number of failures is not directly derived from the parameters of the consecutive-(r, s)-out-of-(m, n) model, so it must be calculated before building the model.

Theorem 1. The maximum allowable number of failures for a consecutive-(r, s)-out-of-(m, n) system is equal to:

$$M = m \times n - \left\lfloor \frac{m}{r} \right\rfloor \times \left\lfloor \frac{n}{s} \right\rfloor. \tag{4}$$

Proving. We will conditionally divide the entire system into $r \times s$ blocks. For the system to remain operational, each $r \times s$ block must contain at least one functioning element. To achieve this, the functioning elements can be placed at intervals of r vertically and s horizontally. As a result, we obtain $\left\lfloor \frac{m}{r} \right\rfloor \times \left\lfloor \frac{n}{s} \right\rfloor$ functioning elements. The total number of elements in the system is $m \times n$. Subtract the number of functioning elements from the total to determine the number of failed elements:

$$m \times n - \left\lfloor \frac{m}{r} \right\rfloor \times \left\lfloor \frac{n}{s} \right\rfloor, \tag{5}$$

and this gives us M – the maximum number of failed elements under which the system remains operational.■

Let's look at an example. Suppose we are given a consecutive-(3, 4)-out-of-(7, 7) system, i.e., $m = n = 7$, $r = 3$, $s = 4$. We will now determine the value of M:

$$M = 7 \times 7 - \left\lfloor \frac{7}{3} \right\rfloor \times \left\lfloor \frac{7}{4} \right\rfloor = 49 - 2 \times 1 = 47. \tag{6}$$

The system can be schematically represented as a 7×7 table, where each cell represents the state of a corresponding processor. Table 1 shows such a system under the condition that all its processors are functioning.

As shown, $M = 47$, meaning that the system under consideration can function even if 47 out of 49 elements fail. Table 2 provides an example of such a situation. As can be seen, the system indeed contains no 3×4 blocks composed entirely of failed elements.

After determining the maximum number of failures under which the system remains operational, we can construct the MLE-model $K(M, m \times n)$, i.e. $K(47, 49)$. The resulting MLE-model is then weakened on all failure blocks of size $r \times s$ (i.e., 3×4)

The number of such blocks can be determined using the formula:

$$c = (m - r + 1) \times (n - s + 1). \tag{7}$$

Theorem 2. To weaken the basic MLE-model on vectors that contain zeros at positions corresponding to a block of $r \times s$ elements, it is sufficient to multiply any two edge functions of the

GL-model by the expression *f′(X)* – the conjunction of all possible disjunctions of elements forming $r \times s$ blocks.

Proving. For each block of processors of size $r \times s$, we construct a disjunction:

$$D_{i,j} = x_{i,j} \vee x_{i,j+1} \vee ... \vee x_{i,j+s} \vee x_{i+1,j} \vee \vee x_{i+1,j+1} \vee ... \vee x_{i+r,1} ... \vee x_{i+r,j+s}; \quad (8)$$

$$D_{i,j} = \vee_{i}^{i+r-1} \vee_{j}^{i+s-1} x_{i,j}.$$

It is clear that $D_{i,j}$ will take the value 0 if and only if all variables representing the states of the corresponding processors take the value 0 — which corresponds to the situation where all these processors have failed.

We combine the disjunctions of all $r \times s$ blocks into a conjunction:

$$f' = D_{1,1} \wedge D_{1,2} \wedge ... \wedge D_{1,m-r+1} \wedge D_{2,1} \wedge \wedge D_{2,2} \wedge ... \wedge D_{2,m-r+1} \wedge ... \wedge D_{n-s+1,1} \wedge \wedge D_{n-s+1,2} \wedge .... \wedge D_{n-s+1,m-r+1}. \quad (9)$$

In other words,

$$f' = \wedge_{p=1}^{m-r+1} \wedge_{q=1}^{n-s+1} D_{i,j} = = \wedge_{p=1}^{m-r+1} \wedge_{q=1}^{n-s+1} \left(\vee_{i=p}^{i+r-1} \vee_{j=q}^{i+s-1} x_{i,j} \right). \quad (10)$$

As can be seen, if all processors in at least one block fail, the conjunction *f′* will take the value 0, and if there are no such blocks, *f′* will take the value 1.

Multiply any two edge functions of the model by the expression *f′*. Thus, if all processors in at least one block fail, *f′* will take the value 0, and consequently, both modified edge functions will also take the value 0, resulting in the exclusion of these two edges from the graph. The loss of at least two edges in a cyclic graph leads to a loss of connectivity in the model’s graph, which accurately corresponds to the system’s failure.

If the system contains no block composed entirely of failed processors, then *f′* = 1, and the modified functions will take the same values as they did before the modification. In this case, the behavior of the model will remain unchanged. ■

It is worth noting that weakening the model by modifying any two of its edge functions, although it changes the model so that it corresponds to a given system, this approach is not always optimal and requires further research.

COMBINED K-OUT-OF-N, CONSECUTIVE-K_c-OUT-OF-N AND CONSECUTIVE-(R, S)-OUT-OF-(M, N) SYSTEMS

The system considered above, although not a basic system – since it can operate with varying numbers of functioning components – still fails when a single condition is met: the failure of a cluster of elements of size $r \times s$. However, real-world systems are often more complex and may fail under multiple conditions.

For example, in [33], the system consists of elements arranged in multiple rows. The system has *m* rows, each containing *n* elements.

The described system fails when any of the following three conditions is met:

1) *k*-out-of-*n* – failure of a fixed number of any elements. Essentially, this condition describes the behavior of a basic system. In [33], an example is given of a system that fails when 10 percent or more of the elements fail;

2) Consecutive-k_c-out-of-*n* – failure of *k* consecutive elements in a single row. That is, the entire system fails if *k* consecutively connected elements fail in any one row;

3) Consecutive-(*r*, *s*)-out-of-(*m*, *n*) – failure of a cluster of elements of size $r \times s$.

It should be noted that in [33] *k*-out-of-*n*:F, consecutive-k_c-out-of-*n*:F, and consecutive-(*r*, *s*)-out-of-(*m*, *n*):F systems type are considered. However, for the sake of brevity, we will refer to these system types simply as *k*-out-of-*n*, consecutive-k_c-out-of-*n*, and consecutive-(*r*, *s*)-out-of-(*m*, *n*), respectively, keeping in mind that all these types describe conditions under which the system fails: failure of any *k* components, failure of k_c consecutive components, or failure of a block of components of size $r \times s$.

Table 1. **A schematic representation of a consecutive-(r, s)-out-of-(m, n) system with** ***m*** **=** ***n*** **= 7;** ***r*** **= 3;** ***s*** **= 4**

1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1

Source**: compiled by the authors**

Table 2. **An example of a maximum failure configuration under which the system remains operational**

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

Source: **compiled by the authors**

Let's start building a GL-model by analyzing the given conditions. First, determine whether the conditions are mutually consistent. It is known that the system fails when k components fail.

Clearly, under this condition, the maximum allowable number of failures is:

$$M = n - k + 1. \quad (11)$$

It is also possible to determine the maximum allowable number of failures M_{RS} under which the system remains operational for the consecutive-(r, s)-out-of-(m, n) condition using formula (4). In addition, [32] describes a method for determining the maximum allowable number of failures for the consecutive-k-out-of-n condition. According to this condition, failures in each row are considered separately, since the system fails when k_c consecutive components fail, and components in different rows cannot be consecutive. Therefore, we calculate the maximum allowable number of failures per row and multiply the resulting number by m – the number of rows in the system. The resulting number, M_C, will be the maximum allowable number of failures under the consecutive-k-out-of-n condition. Thus, the minimum of M, M_C, and M_{RS} will represent the overall maximum allowable number of failures.

It is important to note that the value of k_c must be greater than s from the consecutive-(r, s)-out-of-(m, n) condition, i.e., greater than the number of consecutive failed elements in each row of the block. Otherwise, the consecutive-(r, s)-out-of-(m, n) condition may be completely ignored, since regardless of how many rows of a block contain s failures, the system will fail anyway due to the failure of $k_c \le s$ consecutively connected elements.

After analyzing all system failure conditions and determining the maximum allowable number of failures, we construct the MLE-model. For the consecutive-k_c-out-of-n and consecutive-(r, s)-out-of-(m, n) conditions, we determine the vectors that cause system failure. The MLE-model is then weakened on all identified vectors. The resulting GL-model will correspond to the mixed system.

ALGORITHM

Thus, to summarize the above, the general algorithm for constructing GL-models for systems of the consecutive-(r, s)-out-of-(m, n) type, as well as for mixed systems of the k-out-of-n, consecutive-k-out-of-n, and consecutive-(r, s)-out-of-(m, n) types, is as follows:

1) determine the maximum allowable number of failures M under which the system remains operational. If there is more than one failure condition, M is taken as the minimum among the corresponding values calculated for each condition individually;

2) construct an auxiliary MLE-model of the basic system, using M as the allowable number of failures;

3) identify all failure combinations that cause the system to fail but which are not reflected on MLE-model;

4) weaken the model on the system state vectors corresponding to the failure combinations identified in the previous step.

EXAMPLES

Example 1. Consider a linear consecutive-(r, s)-out-of-(m, n) system. An example of such a system may be a system composed of sensors and processors. The processors are arranged in a matrix with m rows and n columns. Each sensor is connected to a block of processors of size $r \times s$. Each processor is connected to an external node as well as to all neighboring processors. A processor may be connected to multiple sensors. In the reliability analysis of the system, external nodes will not be taken into account (as they introduce additional failure conditions). Thus, the system fails when all elements in a block of size $r \times s$ fail.

Suppose a consecutive-(r, s)-out-of-(m, n) is given, where $m = 3$, $n = 4$, $r = 2$, $s = 2$ (Fig. 1).

Let's find the maximum allowable number of failures using formula (4):

$$M = 3 \times 4 - \left\lfloor \frac{3}{2} \right\rfloor \times \left\lfloor \frac{4}{2} \right\rfloor = 12 - 1 \times 2 = 10. \quad (12)$$

For example, the system will remain operational under the failure configuration shown in Table 3. Let's construct MLE-model $K(10, 12)$.

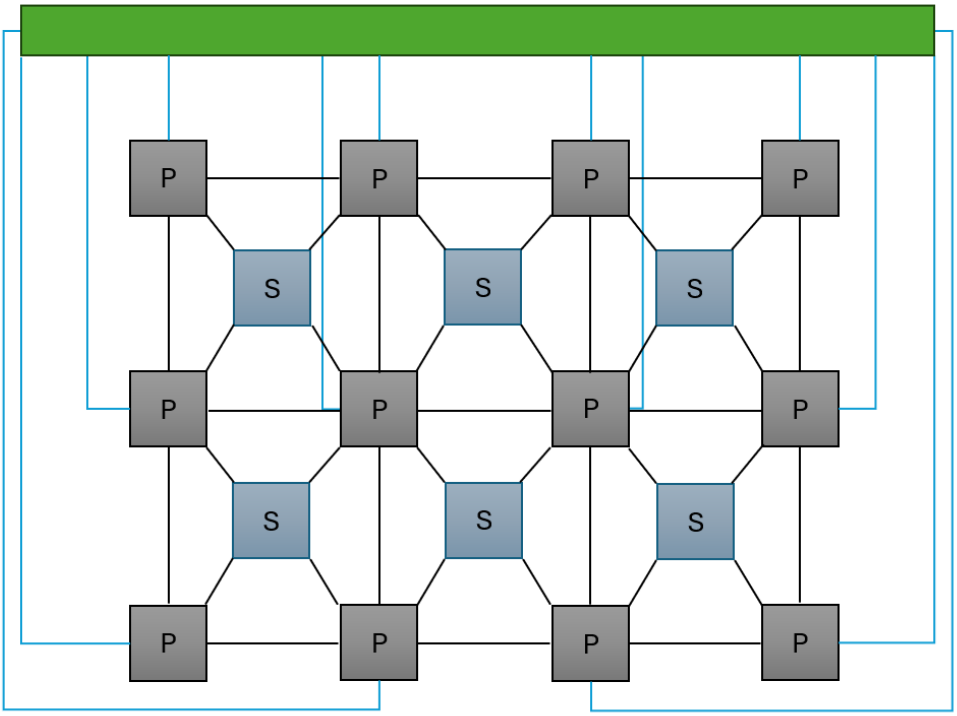

Fig. 1. **An example of a consecutive-(*r*, *s*)-out-of-(*m*, *n*) system with *m* = 3; *n* = 4; *r* = 2; *s* = 2**

Source: **compiled by the authors**

Define the edge functions and the graph structure (Fig. 2):

$$f_1 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6 \vee (x_7 \vee \vee x_8 \vee x_9 \vee x_{10}x_{11}x_{12})((x_7 \vee x_8)(x_7x_8 \vee \vee x_9) \vee (x_{10} \vee x_{11})(x_{10}x_{11} \vee x_{12}))(x_7x_8x_9 \vee \vee x_{10} \vee x_{11} \vee x_{12});$$

$$f_2 = (x_1 \vee x_2 \vee x_3 \vee (x_4 \vee x_5)(x_4x_5 \vee x_6)) \wedge \wedge((x_1 \vee x_2)(x_1x_2 \vee x_3) \vee x_4 \vee x_5 \vee x_6) \vee \vee(x_7 \vee x_8 \vee x_9 \vee (x_{10} \vee x_{11})(x_{10}x_{11} \vee x_{12})) \wedge \wedge((x_7 \vee x_8)(x_7x_8 \vee x_9) \vee x_{10} \vee x_{11} \vee x_{12});$$

$$f_3 = (x_1 \vee x_2 \vee x_3 \vee x_4x_5x_6)((x_1 \vee x_2) \wedge \wedge(x_1x_2 \vee x_3) \vee (x_4 \vee x_5)(x_6x_7 \vee x_8)) \wedge \wedge(x_1x_2x_3 \vee x_4 \vee x_5 \vee x_6) \vee x_7 \vee x_8 \vee x_9 \vee \vee x_{10} \vee x_{11} \vee x_{12}.$$

Table 3. **An example of a maximum failure configuration under which the system remains operational**

x_1	x_2	x_3	x_4
x_5	x_6	x_7	x_8
x_9	x_{10}	x_{11}	x_{12}

Source: **compiled by the authors**

As the next step, we modify the GL-model to take into account the system failure condition due to the failure of a block of elements of size $r \times s$. To do this, we weaken the obtained model on the vectors corresponding to all possible combinations of failure blocks of size $r \times s$. According to (7), the number of such combinations is:

$$c = (3-2+1) \times (4-2+1) = 6. \qquad (13)$$

The above mentioned vectors are shown in Table 4.

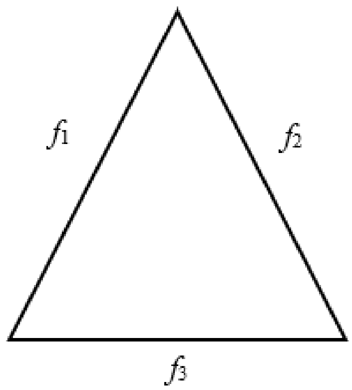

Fig. 2. **The graph of the GL-model *K*(10, 12)**

Source: **compiled by the authors**

In this example, the model will be weakened by modifying the expressions of the edge functions. The function f' represents all possible combinations of $r \times s$ blocks that cause the system to fail:

$$f' = (x_1 \vee x_2 \vee x_5 \vee x_6)(x_2 \vee x_3 \vee x_6 \vee x_7) \wedge$$
$$\wedge (x_3 \vee x_4 \vee x_7 \vee x_8)(x_5 \vee x_6 \vee x_9 \vee x_{10}) \wedge$$
$$\wedge (x_6 \vee x_7 \vee x_{10} \vee x_{11})(x_7 \vee x_8 \vee x_{11} \vee x_{12}).$$

Table 4. **All vectors corresponding to a failed block of elements of size *r*×*s***

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
0	**0**	1	1	**0**	**0**	1	1	1	1	1	1
1	**0**	**0**	1	1	**0**	**0**	1	1	1	1	1
1	1	**0**	**0**	1	1	**0**	**0**	1	1	1	1
1	1	1	1	**0**	**0**	1	1	**0**	**0**	1	1
1	1	1	1	1	**0**	**0**	1	1	**0**	**0**	1
1	1	1	1	1	1	**0**	**0**	1	1	**0**	**0**

Source: **compiled by the authors**

Modify the functions f_1 and f_2 by multiplying their expressions by f':

$$f_1' = (x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6 \vee (x_7 \vee$$
$$\vee x_8 \vee x_9 \vee x_{10}x_{11}x_{12})((x_7 \vee x_8)(x_7x_8 \vee$$
$$\vee x_9) \vee (x_{10} \vee x_{11})(x_{10}x_{11} \vee x_{12}))(x_7x_8x_9 \vee$$
$$\vee x_{10} \vee x_{11} \vee x_{12}))((x_1 \vee x_2 \vee x_5 \vee x_6)(x_2 \vee$$
$$\vee x_3 \vee x_6 \vee x_7) \wedge (x_3 \vee x_4 \vee x_7 \vee x_8)(x_5 \vee$$
$$\vee x_6 \vee x_9 \vee x_{10}) \wedge (x_6 \vee x_7 \vee x_{10} \vee x_{11})(x_7 \vee$$
$$\vee x_8 \vee x_{11} \vee x_{12}));$$

$$f_2' = ((x_1 \vee x_2 \vee x_3 \vee (x_4 \vee x_5)(x_4x_5 \vee x_6)) \wedge$$
$$\wedge ((x_1 \vee x_2)(x_1x_2 \vee x_3) \vee x_4 \vee x_5 \vee x_6) \vee$$
$$\vee (x_7 \vee x_8 \vee x_9 \vee (x_{10} \vee x_{11})(x_{10}x_{11} \vee x_{12})) \wedge$$
$$\wedge ((x_7 \vee x_8)(x_7x_8 \vee x_9) \vee x_{10} \vee x_{11} \vee x_{12})) \wedge$$
$$\wedge ((x_1 \vee x_2 \vee x_5 \vee x_6)(x_2 \vee x_3 \vee x_6 \vee x_7) \wedge$$
$$\wedge (x_3 \vee x_4 \vee x_7 \vee x_8) \wedge (x_5 \vee x_6 \vee x_9 \vee x_{10}) \wedge$$
$$\wedge (x_6 \vee x_7 \vee x_{10} \vee x_{11}) \wedge (x_7 \vee x_8 \vee x_{11} \vee x_{12})).$$

Let's make sure that the resulting model corresponds to the given system. For example, the graph loses connectivity on vectors 001100111111, 111111001100, 111100110011, 110011001111. At the same time, connectivity is preserved on such vectors as 000001010000, 000011110000, and 010101010101.

Example 2. Consider a mixed system that simultaneously satisfies three conditions: *k*-out-of-*n*, consecutive-k_c-out-of-*n* and consecutive-(*r*, *s*)-out-of-(*m*, *n*) (Fig. 3).

Let's take the FTMS from the previous example as a basis and modify it so that it also satisfies the other two conditions. To meet the consecutive-k_c-out-of-*n* condition, we connect each row of processors to additional nodes in such a way that connectivity between the nodes is lost when three consecutively connected processors fail, i.e., $k_c = 3$. From the previous example, we recall that $m = 3$,

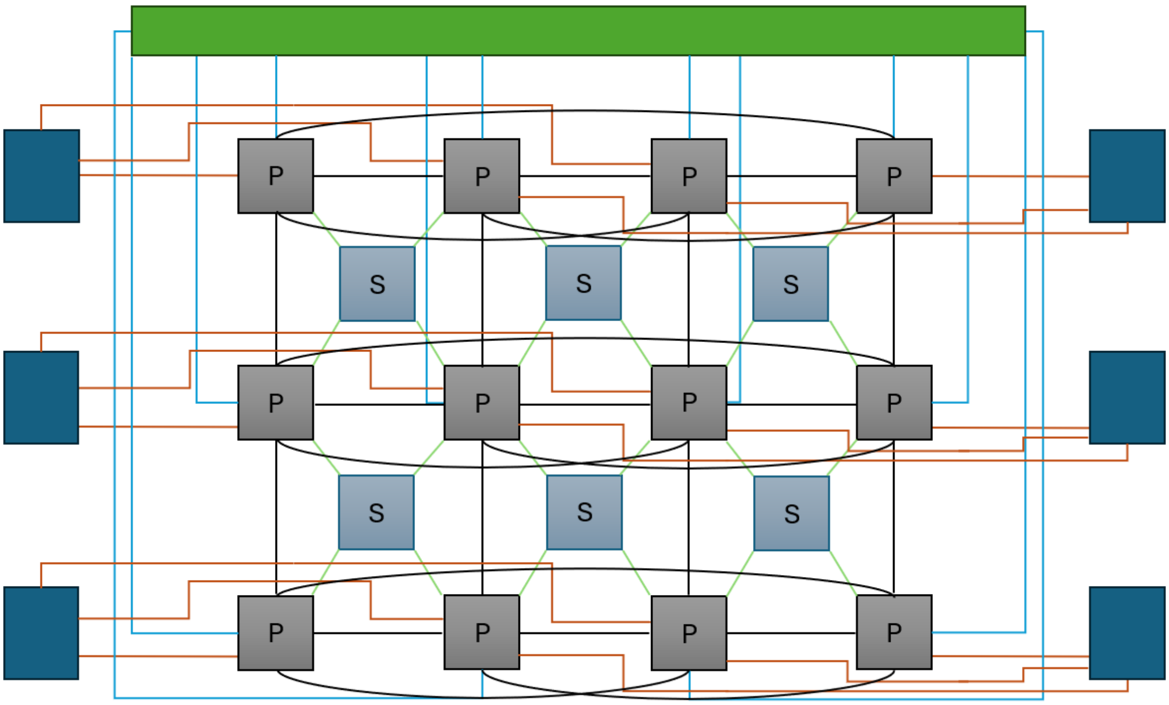

Fig. 3. **An example of a mixed *k*-out-of-*n*, consecutive-k_c-out-of-*n*, and consecutive-(*r*, *s*)-out-of-(*m*, *n*) system with $k = 7$; $k_c = 3$; $m = 3$; $n = 4$; $r = 2$; $s = 2$**

Source: **compiled by the authors**

$n = 4$, $r = 2$, $s = 2$. Thus, since $s < k_c$, both conditions must be taken into account.

For the k-out-of-n condition, the structure does not need to be changed. We only specify that the system fails, for example, when any 7 processors fail. In other words, $k = 7$. Let us determine the minimum allowable number of failures.

Recall that $k = 7$, therefore:

$$M = n - k + 1 = 12 - 7 + 1 = 6. \quad (14)$$

For the consecutive-(r, s)-out-of-(m, n) condition, M_{RS} was determined in the previous example and is equal to 10.

Using the method proposed in [32], we determine the Using the method proposed in [32], we determine the maximum allowable number of failures in a single row under which the system remains operational:

$$M_r = n_r - \left\lfloor \frac{n_r}{k_c} \right\rfloor = 4 - \left\lfloor \frac{4}{3} \right\rfloor = 3. \quad (15)$$

Let's compare the maximum allowable number of failures for the different conditions:

$$M < M_C < M_{RS}. \quad (16)$$

Thus, the system fails when any $M + 1$ elements fail, so we construct the MLE-model $K(M, n)$, or $K(6, 12)$.

Accordingly, M_C and M_{RS} are not considered in further calculations. For $K(6, 12)$, we obtain the following edge functions:

$$f_1 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6;$$

$$f_2 = (x_1 \vee x_2 \vee x_3 \vee (x_4 \vee x_5)(x_4x_5 \vee x_6)) \wedge \wedge((x_1 \vee x_2)(x_1x_2 \vee x_3) \vee x_4 \vee x_5 \vee x_6) \vee \vee x_7x_8x_9x_{10}x_{11}x_{12};$$

$$f_3 = ((x_1 \vee x_2 \vee x_3 \vee x_4x_5x_6)((x_1 \vee x_2) \wedge \wedge(x_1x_2 \vee x_3) \vee (x_4 \vee x_5)(x_6x_7 \vee x_8)) \wedge \wedge(x_1x_2x_3 \vee x_4 \vee x_5 \vee x_6)) \vee ((x_7 \vee x_8) \wedge \wedge(x_7x_8 \vee x_9)(x_7x_8x_9 \vee x_{10}x_{11}x_{12}) \wedge \wedge(x_{10} \vee x_{11})(x_{10}x_{11} \vee x_{12}));$$

$$f_4 = (x_1 \vee x_2 \vee x_3)((x_1 \vee x_2)(x_1x_2 \vee x_3) \vee \vee x_4x_5x_6)(x_1x_2x_3 \vee (x_4 \vee x_5)(x_4x_5 \vee x_6)) \wedge \wedge(x_4 \vee x_5 \vee x_6) \vee (x_7 \vee x_8 \vee x_9)((x_7 \vee x_8) \wedge \wedge(x_7x_8 \vee x_9) \vee x_{10}x_{11}x_{12})(x_7x_8x_9 \vee \vee(x_{10} \vee x_{11})(x_{10}x_{11} \vee x_{12}))(x_{10} \vee x_{11} \vee x_{12});$$

$$f_5 = ((x_1 \vee x_2)(x_1x_2 \vee x_3)(x_1x_2x_3 \vee x_4x_5x_6) \wedge (x_4 \vee x_5)(x_4x_5 \vee x_6)) \vee ((x_7 \vee x_8 \vee x_9 \vee \vee x_{10}x_{11}x_{12})((x_7 \vee x_8)(x_7x_8 \vee x_9) \vee \vee(x_{10} \vee x_{11})(x_{10}x_{11} \vee x_{12})) \vee (x_7x_8x_9 \vee x_{10} \vee x_{11} \vee x_{12}))$$

$$f_6 = x_1x_2x_3x_4x_5x_6 \vee \vee(x_7 \vee x_8 \vee x_9 \vee (x_{10} \vee x_{11})(x_{10}x_{11} \vee x_{12})) \wedge \wedge((x_7 \vee x_8)(x_7x_8 \vee x_9) \vee x_{10} \vee x_{11} \vee x_{12});$$

$$f_7 = x_7 \vee x_8 \vee x_9 \vee x_{10} \vee x_{11} \vee x_{12}.$$

According to [32], let's determine the number of vectors with three consecutive failures in a single row that lead to system failure:

$$c_r = n_r - k_c + 1 = 4 - 3 + 1 = 2. \quad (17)$$

The total number of such vectors is:

$$c = c_r \times m = 2 \times 3 = 6. \quad (18)$$

Table 5 lists all vectors containing k_c consecutive failures.

We weaken the MLE-model on these vectors by modifying the edge functions f_1 and f_7:

$$f_1' = (x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6) \wedge \wedge(x_1 \vee x_2 \vee x_3)(x_2 \vee x_3 \vee x_4)(x_5 \vee x_6 \vee x_7) \wedge \wedge(x_6 \vee x_7 \vee x_8)(x_9 \vee x_{10} \vee x_{11}) \wedge \wedge(x_{10} \vee x_{11} \vee x_{12});$$

$$f_7' = (x_7 \vee x_8 \vee x_9 \vee x_{10} \vee x_{11} \vee x_{12}) \wedge (x_1 \vee x_2 \vee x_3)(x_2 \vee x_3 \vee x_4)(x_5 \vee x_6 \vee x_7) \wedge \wedge(x_6 \vee x_7 \vee x_8)(x_9 \vee x_{10} \vee x_{11}) \wedge \wedge(x_{10} \vee x_{11} \vee x_{12}).$$

The modified model $K'(6, 12)$ is obtained.

Table 5. **All vectors with three consecutive failures**

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
0	0	0	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	1
1	1	1	1	1	1	1	1	1	0	0	0

Source: **compiled by the authors**

To satisfy the consecutive-(r, s)-out-of-(m, n) condition, we will weaken $K'(6, 12)$ on the vectors defined in the previous example (Table 4). To do this, we modify the functions f_2 and f_6. The resulting model, $K''(6, 12)$, will have the following edge functions:

$$f_1^{'} = (x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6) \wedge$$
$$\wedge(x_1 \vee x_2 \vee x_3)(x_2 \vee x_3 \vee x_4)(x_5 \vee x_6 \vee x_7) \wedge$$
$$\wedge(x_6 \vee x_7 \vee x_8)(x_9 \vee x_{10} \vee x_{11}) \wedge$$
$$\wedge(x_{10} \vee x_{11} \vee x_{12});$$
$$f_2^{'} = ((x_1 \vee x_2 \vee x_3 \vee (x_4 \vee x_5)(x_4 x_5 \vee x_6)) \wedge$$
$$\wedge((x_1 \vee x_2)(x_1 x_2 \vee x_3) \vee x_4 \vee x_5 \vee x_6) \vee$$
$$\vee x_7 x_8 x_9 x_{10} x_{11} x_{12}))((x_1 \vee x_2 \vee x_5 \vee x_6)(x_2 \vee$$
$$\vee x_3 \vee x_6 \vee x_7) \wedge (x_3 \vee x_4 \vee x_7 \vee x_8)(x_5 \vee$$
$$\vee x_6 \vee x_9 \vee x_{10}) \wedge (x_6 \vee x_7 \vee x_{10} \vee x_{11})(x_7 \vee$$
$$\vee x_8 \vee x_{11} \vee x_{12}));$$
$$f_3 = ((x_1 \vee x_2 \vee x_3 \vee x_4 x_5 x_6)((x_1 \vee x_2) \wedge$$
$$\wedge(x_1 x_2 \vee x_3) \vee (x_4 \vee x_5)(x_4 x_5 \vee x_6)) \wedge$$
$$\wedge(x_1 x_2 x_3 \vee x_4 \vee x_5 \vee x_6)) \vee ((x_7 \vee x_8) \wedge$$
$$\wedge(x_7 x_8 \vee x_9)(x_7 x_8 x_9 \vee x_{10} x_{11} x_{12}) \wedge$$
$$\wedge(x_{10} \vee x_{11})(x_{10} x_{11} \vee x_{12}));$$
$$f_4 = (x_1 \vee x_2 \vee x_3)((x_1 \vee x_2)(x_1 x_2 \vee x_3) \vee$$
$$\vee x_4 x_5 x_6)(x_1 x_2 x_3 \vee (x_4 \vee x_5)(x_4 x_5 \vee x_6)) \wedge$$
$$\wedge(x_4 \vee x_5 \vee x_6) \vee (x_7 \vee x_8 \vee x_9)((x_7 \vee x_8) \wedge$$
$$\wedge(x_7 x_8 \vee x_9) \vee x_{10} x_{11} x_{12})(x_7 x_8 x_9 \vee$$
$$\vee(x_{10} \vee x_{11})(x_{10} x_{11} \vee x_{12}))(x_{10} \vee x_{11} \vee x_{12});$$
$$f_5 = ((x_1 \vee x_2)(x_1 x_2 \vee x_3)(x_1 x_2 x_3 \vee x_4 x_5 x_6) \wedge$$
$$(x_4 \vee x_5)(x_4 x_5 \vee x_6)) \vee ((x_7 \vee x_8 \vee x_9 \vee$$
$$\vee x_{10} x_{11} x_{12})((x_7 \vee x_8)(x_7 x_8 \vee x_9) \vee$$
$$\vee(x_{10} \vee x_{11})(x_{10} x_{11} \vee x_{12})) \wedge$$
$$(x_7 x_8 x_9 \vee x_{10} \vee x_{11} \vee x_{12}))$$
$$f_6^{'} = (x_1 x_2 x_3 x_4 x_5 x_6 \vee$$
$$\vee(x_7 \vee x_8 \vee x_9 \vee (x_{10} \vee x_{11})(x_{10} x_{11} \vee x_{12})) \wedge$$
$$\wedge((x_7 \vee x_8)(x_7 x_8 \vee x_9) \vee x_{10} \vee x_{11} \vee x_{12})) \wedge$$
$$\wedge((x_1 \vee x_2 \vee x_5 \vee x_6)(x_2 \vee x_3 \vee x_6 \vee x_7) \wedge$$
$$\wedge(x_3 \vee x_4 \vee x_7 \vee x_8)(x_5 \vee \vee x_6 \vee x_9 \vee x_{10}) \wedge$$
$$\wedge(x_6 \vee x_7 \vee x_{10} \vee x_{11})(x_7 \vee x_8 \vee x_{11} \vee x_{12}));$$
$$f_7^{'} = (x_7 \vee x_8 \vee x_9 \vee x_{10} \vee x_{11} \vee x_{12}) \wedge$$
$$(x_1 \vee x_2 \vee x_3)(x_2 \vee x_3 \vee x_4)(x_5 \vee x_6 \vee x_7) \wedge$$
$$\wedge(x_6 \vee x_7 \vee x_8)(x_9 \vee x_{10} \vee x_{11}) \wedge$$
$$\wedge(x_{10} \vee x_{11} \vee x_{12}).$$

Let us verify that the resulting model satisfies the specified conditions. For instance, its graph loses connectivity on vectors containing 7 or more zeros: 001001101001, 010001010100, 010100101001, 010110100100. The graph also loses connectivity on vectors containing 3 zeros at positions corresponding to consecutive elements within a single row: 000111111111, 111110001111, 111111111000. Additionally, it loses connectivity if the zeros are located at positions corresponding to a block of elements of size 2×2: 111111001100, 100110011111. At the same time, the graph retains connectivity on vectors such as 110110011101 and 010101010101.

COMPARISON WITH CONVENTIONAL METHODS FOR CONSTRUCTING GL-MODEL

Since no universal method existed for constructing GL-models for such systems – aside from the one described above – each model had to be developed individually, with the approach varying depending on the system type and parameter values.

Let us consider the system from the example to compare the default and proposed methods. Suppose a given system consecutive-(2, 2)-out-of-(3, 4). According to the conventional approach we begin by constructing a basic model. The system consists of 3 · 4 = 12 elements in total and can fail when at least 2 · 2 = 4 elements fail (in other words it is tolerant to 3 arbitrary failures). Therefore, we construct the basic MLE-model $K(3, 12)$.

According to [35]:

$$\rho(m,n) = 12 - 3 + 1 = 10. \qquad (19)$$

Such a model contains 10 edges and 10 edge functions, respectively.

However, the system does not fail if 4 or more failed elements do not form a block of size $r \times s$ (i.e. 2×2). Thus, the model must be enhanced so that its graph remains connected for input vectors containing 4 or more zeros, provided that corresponding failed processors do not form a contiguous 2×2 block. The total number of binary vectors containing from 4 to 10 zeros (vectors with 11 or 12 zeros are not considered, as it is straightforward to show that the system is guaranteed to fail in all such configuration) is:

$$\sum_{k=4}^{10} \binom{12}{k} = 3784. \qquad (20)$$

Using an exhaustive enumeration script written in Python, it was determined that 1051 of these vectors contain at least one contiguous block of failed elements of size 2×2.

Therefore, the number of configurations with 4 or more failures that do not lead to system failure is:

$$3784 - 1051 = 2733. \quad (21)$$

These 2733 configurations represent failure scenarios in which the system remains operational due to the spatial distribution of the failed elements, i.e., no critical 2×2 failure block is formed.

The next step is to strengthen basic model across all 2733 configurations. This means modifying the model so that its graph maintains connectivity under each of these failure scenarios.

It is worth recalling that, in the proposed method, model consists only 3 edges (i.e., 70% fewer). Also, the model must be modified by weakening on just 6 vectors instead of strengthening on 2733 vectors (representing a 99.78% reduction in the number of cases to be handled).

In general, the efficiency of the proposed method compared to the conventional approach may differ from the results obtained in this example and depends on the specific configuration of the system for which the GL-model is being constructed.

CONCLUSIONS

The paper proposes methods for constructing GL-models for consecutive-(r, s)-out-of-(m, n) systems, as well as for mixed-type systems such as k-out-of-n, consecutive-k-out-of-n, and consecutive-(r, s)-out-of-(m, n). In consecutive-(r, s)-out-of-(m, n) systems, processors are arranged in the form of a rectangular matrix, and the entire system fails if a rectangular block of elements of size r×s fails, unlike basic k-out-of-n systems, which fail when any k components fail. Mixed systems have more than one failure condition. Three cases of such conditions are considered: failure of a fixed number of any k components; failure of k_c consecutively connected components in a single row; and failure of a block of components of size r×s.

The paper presents a universal method for calculating the maximum possible number of failed elements under which the system remains operational, both for consecutive-(r, s)-out-of-(m, n) systems and for the mixed k-out-of-n, consecutive-k-out-of-n, and consecutive-(r, s)-out-of-(m, n) systems. For the consecutive-(r, s)-out-of-(m, n) system, the method reduces to determining the minimal number of functioning processors required for the system to remain operational. For a mixed system, the maximum allowable number of failed components is determined separately for each failure condition. The smallest of the obtained values represents the maximum number of component failures under which the system can still function.

An algorithm for constructing GL-models for both types of systems is presented. In the general case, any number of failure conditions can be considered, and the proposed approach can be applied to all of them in a similar manner.

The methods of constructing GL-models of both system types are based on MLE-models (minimum lost edges). Knowing the maximum possible number of failures under which the system remains operational, it is proposed to first build a basic MLE-model for this failure count. The resulting MLE-model is then weakened on the vectors corresponding to failure configurations that lead to system failure. MLE-models can be weakened either by modifying the edge functions, altering the graph structure, or combining both approaches.

The paper demonstrates examples of GL-model construction for consecutive-(r, s)-out-of-(m, n) and mixed systems. Experiments have been conducted to confirm that the obtained models correspond to the behavior of the given systems in the failure flows.

The proposed approach can be applied not only to fault-tolerant systems where the elements are processors, but also to systems composed of other types of elements.

The proposed model construction method broadens the applicability of GL-models to previously unaddressed classes of systems. GL-models provide a flexible framework for representing system behavior under failure flow involving heterogeneous subsystems with varying failure criteria. Additional failure conditions can be seamlessly integrated into the GL-model. In contrast, accommodating such conditions within traditional analytical approaches to system reliability assessment often necessitates substantial recalculations or methodological modifications.

Future research may focus on alternative ways of modifying basic MLE-models, particularly by changing the graph structure, as well as on constructing GL-models for other types of non-basic systems.

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. Peiravi, A., Nourelfath, M. & Kazemi Zanjani, M. "Redundancy strategies assessment and optimization of k-out-of-n systems based on Markov chains and genetic algorithms". *Reliability Engineering & System Safety*. 2022; 221: 108277. DOI: https://doi.org/10.1016/j.ress.2021.108277.

4. Huynh, K. T., Vu, H. C., Nguyen, T. D. & Ho, A. C. "A predictive maintenance model for k-out-of-n:F continuously deteriorating systems subject to stochastic and economic dependencies". *Reliability Engineering & System Safety*. 2022; 226: 108671. DOI: https://doi.org/10.1016/j.ress.2022.108671.

5. Asadi, M. "On the phase transition of k-out-of-n systems with applications to optimal maintenance". *Journal of Computational and Applied Mathematics*. 2024; 435: 115286. DOI: https://doi.org/10.1016/j.cam.2023.115286.

6. Dui, H., Tian, T., Zhao, J. & Wu, S. "Comparing with the joint importance under consideration of consecutive-k-out-of-n system structure changes". *Reliability Engineering & System Safety*. 2022; 219: 108255. DOI: https://doi.org/10.1016/j.ress.2021.108255.

7. Wang, Y., Hu, L., Yang, L. & Li, J. "Reliability modeling and analysis for linear consecutive-k-out-of-n: F retrial systems with two maintenance activities". *Reliability Engineering & System Safety*. 2022; 226: 108665. DOI: https://doi.org/10.1016/j.ress.2022.108665.

8. Rezaei, E., Jafary, B. & Fiondella, L. "Optimal maintenance policies for linear consecutive k-out-of-n:F systems susceptible to dependent failures". *Computers & Industrial Engineering*. 2022; 173: 108657. DOI: https://doi.org/10.1016/j.cie.2022.108657.

9. 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.

10. 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.

11. 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.

12. Chachra, A., Ram, M. & Kumar, A. "A pythagorean fuzzy approach to consecutive k-out-of-r-from-n system reliability modelling". *International Journal of System Assurance Engineering and Management*. 2024. p. 1–12, DOI: https://doi.org/10.1109/TR.2004.833313.

13. 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.

14. Wu, C., Zhao, X., Wang, S. & Song, Y. "Reliability analysis of consecutive-k-out-of-r-from-n subsystems: F balanced systems with load sharing". *Reliability Engineering & System Safety*. 2022; 228: 108776. DOI: https://doi.org/10.1016/j.ress.2022.108776.

15. 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.

16. 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.

17. 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.

18. Triantafyllou, I. S. “On the combined m-consecutive-k-out-of-n: F and consecutive k_c-out-of-n: F reliability system: Some advances”. *Advances in Reliability Science 4.0*. 2023. p. 463–476. DOI: https://doi.org/10.1016/B978-0-323-99204-6.00004-2.

19. 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

20. 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.

21. Cui, L., Wang, M. & Jiang, W. “Reliability analysis of a combination of (n,f,k) and ⟨n,f,k⟩ systems”. *Reliability Engineering & System Safety*. 2024; 249: 110191. DOI: https://doi.org/10.1016/j.ress.2024.110191.

22. 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.

23. Kan C. & Ozkut M. “Analysis of joint reliability importance in linear m-consecutive-k, l-out-of-n: F system”. *Communications Faculty of Sciences University of Ankara Series A1 Mathematics and Statistics*. 2020; 69 (1): 699–716. DOI: https://doi.org/10.31801/cfsuasmas.635829.

24. Özbey, F. “Reliability evaluation of m-consecutive-k,l-out-of-n:F system subjected to shocks”. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. 2022; 236 (6): 1135–1146, https://www.scopus.com/record/display.uri?eid=2-s2.0-85116069008&origin=resultslist. DOI: https://doi.org/10.1177/1748006X211048992.

25. Zhang, C., Yang, J., Li, M. & Wang, N. “Reliability analysis of a two-dimensional linear consecutive-(*r*, *s*)-out-of-(*m*, *n*):F repairable system”. *Reliability Engineering & System Safety*. 2024; 242: 109792. DOI: https://doi.org/10.1016/j.ress.2023.109792.

26. Yamada, K., Nakamura, T. & Yamamoto, H. “Generalization of Limit Theorems for Connected-(*r*, *s*)-out-of-(*m*, *n*):F lattice systems”. *IEICE Transactions on Fundamentals*. 2022; E105-A (3): 562–570. DOI: https://doi.org/10.1587/transfun.2020EAP1145.

27. 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.

28. 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.

29. 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.

30. Romankevich, V. A., Rabh Moh’d Ahmad Al Shbul & Nazarenko, V. V. “On the minimization of basic cyclic GL-models” (in Russian). *Visnyk TUP. Technical Sciences*. Khmelnytskyi, Ukraine. 2004; 2 (1): 42–46.

31. 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.

32. Romankevich, V. A., Yermolenko,, I. A., Morozov K. V. & Romankevich A. M. “Graph-logical models for (n, f, k)– and consecutive-k-out-of-n–systems”. *Herald of Advanced Information Technology*. 2024; 7 (3): 296–308. DOI: https://doi.org/10.15276/hait.07.2024.21

33. Zuo, M., Lin, D. & Wu, Y. “Reliability evaluation of combined k-out-of-n: F, consecutive-k-out-of-n: F and linear connected-(r, s)-out-of-(m, n): F system structures”. *IEEE Trans. Reliab.* 49 (2000): 99–104.

34. 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.

35. Romankevich, A. M., Romankevich, V. A. & Maidanyuk, I. V. “Boundary estimates of the number of edges in GL-models of fault-tolerant multiprocessor systems in a failure flow” (in Russian). *Electronic Modeling*. 2008; 30 (1): 59–70.

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 14.04.2025
Received after revision 12.06.2025
Accepted 17.06.2025

DOI: https://doi.org/10.15276/aait.08.2025.15
УДК 004.05

Узагальнення методу побудови GL-моделей складних відмовостійких багатопроцесорних систем з додатковими умовами відмов

Романкевич Віталій Олексійович[1)]
ORCID: https://orcid.org/0000-0003-4696-5935; zavkaf@scs.kpi.ua. Scopus Author ID: 57193263058
Єрмоленко Ігор Андрійович[1)]
ORCID: https://orcid.org/0009-0008-5298-4888; yermolenkomail@gmail.com
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
Мельник Олександр Олегович[1)]
ORCID: https://orcid.org/0009-0003-7527-8146; inboxnumberzero@zoho.com
[1)] National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, 37, Peremogy Ave. Kyiv, 03056, Ukraine

АНОТАЦІЯ

Стаття присвячена методам побудови GL-моделей відмовостійких багатопроцесорних систем. GL-моделі можуть використовуватися як моделі поведінки таких систем у потоці відмов для оцінки їхніх показників надійності шляхом проведення статистичних експериментів. У роботі розглянуто два типи систем: послідовні двовимірні системи та системи змішаного типу. Послідовна двовимірна система – це система, у якій компоненти розташовані у вигляді прямокутної матриці, і яка виходить з ладу при появі прямокутного блоку певного розміру, що містить лише несправні компоненти. Система змішаного типу виходить з ладу, якщо виконується хоча б одна з наступних умов: відмовила задана кількість довільних компонентів; відмовила задана кількість послідовних компонентів; або у прямокутній матриці компонентів з'явився прямокутний блок певного розміру, який складається лише з несправних компонентів. На сьогодні відсутні формалізовані методи побудови GL-моделей для зазначених типів систем. Метою даної роботи є створення універсального методу побудови GL-моделей як для послідовних двовимірних систем, так і для систем змішаного типу. Показано, що для побудови GL-моделі такої системи достатньо визначити максимальну кількість несправних компонентів, за якої система зберігає роботоздатність. На основі цього значення формується базова модель системи без урахування додаткових умов відмови. Далі визначаються всі комбінації відмов компонентів, що призводять до відмови системи. Базова модель послаблюється на векторах, які відповідають цим критичним комбінаціям. У роботі вперше представлено алгоритм побудови GL-моделей для послідовних двовимірних систем та систем змішаного типу. Окрім того, запропоновано методи розрахунку максимально допустимої кількості відмов компонентів, за якої система залишається роботоздатною, а також оцінки загальної кількості комбінацій відмов компонентів, що призводять до її відмови. Результати експериментів підтверджують, що запропоновані моделі адекватно відображають реальну поведінку систем у потоці відмов. Наведено приклади, що ілюструють процес побудови GL-моделей для систем обох вищезазначених типів.

Ключові слова: GL-моделі; небазові відмовостійкі багатопроцесорні системи; послідовні двовимірні системи; системи змішаного типу; оцінка надійності

ABOUT THE AUTHORS

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

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

Ihor A. Yermolenko – 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/0009-0008-5298-4888; yermolenkomail@gmail.com
Research field: GL-models; Fault-tolerant multiprocessor systems reliability estimation

Єрмоленко Ігор Андрійович – аспірант, кафедра Системного програмування і спеціалізованих комп’ютерних систем. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна

Kostiantyn V. Morozov – Candidate of Engineering Sciences, Assistant, System Programming and Specialized Computer System Department. National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, 37, Peremogy Ave. Kyiv, 03056, Ukraine
ORCID: https://orcid.org/0000-0003-0978-6292, mcng@ukr.net, Scopus Author ID: 57222509251
Research field: GL-models; Fault-tolerant multiprocessor systems reliability estimation; Self-diagnosable multiprocessor systems

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

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

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

Oleksandr O. Melnyk – 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/0009-0003-7527-8146; inboxnumberzero@zoho.com
Research field: Fault-tolerant multiprocessor systems reliability estimation; Self-diagnosable multiprocessor systems

Мельник Олександр Олегович – аспірант, кафедра Системного програмування і спеціалізованих комп’ютерних систем. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна