

Miroslav Kvassay - Jozef Kostolny \*

---

## EVALUATION OF ALGORITHMS FOR IDENTIFICATION OF MINIMAL CUT VECTORS AND MINIMAL PATH VECTORS IN MULTI-STATE SYSTEMS

*Minimal Cut Vectors (MCVs) and Minimal Path Vectors (MPVs) are one of the key concepts of reliability analysis. They allow us to estimate system availability or to analyze influence of individual system components on the entire system. However, the main problem of their use, especially in reliability analysis of complex systems, lies in their identification. Several algorithms have been proposed to solve this task. Some of the most universal ones are based on logical differential calculus. These algorithms use integrated direct partial logic derivatives to find situations that can correspond to the MCVs (MPVs) and a special type of logic conjunction to select only those situations that really agree with the MCVs (MPVs). In this paper, we summarize the ideas behind these algorithms in more formal way and present results of some experiments performed to study their time complexity.*

**Keywords:** Reliability, multi-state system, minimal cut vector, minimal path vector, direct partial logic derivative, integrated direct partial logic derivative.

### 1. Introduction

Reliability has been considered as an important characteristic of many systems [1 - 6]. One of the current issues of reliability engineering is the analysis of complex systems [7]. These systems are characterized by the fact that they consist of many elements (components) with very different behavior. Typical instances of such systems are healthcare systems [5] or gas distribution networks [6]. A healthcare system consists of very variable components that can be identified as hardware, software, organizational, and human, while a gas network is composed of many different hardware components, such as pipelines with various capacities, compressor stations, and supply and demand centers. This variability implies that such systems can operate at several performance levels. However, many approaches of reliability engineering are based on the assumption that the system and all its components can be either functioning or failed [8]. This indicates that many methods of reliability analysis have originally been developed for so-called Binary-State Systems (BSSs) and, therefore, they are not very appropriate for the analysis of complex systems operating at several performance levels. Furthermore, classical approaches of reliability analysis assume that the system and all its components have very similar behavior and, therefore, they are not very suitable for the analysis

of complex systems consisting of components with various natures. Because of that, new methods have to be developed for the analysis of such systems. One of the possible ways is application of models that allow defining more than two states in system/components performance. These models are known as Multi-State Systems (MSSs) [7, 9 and 10].

One of the most important tasks of reliability analysis based on BSSs is finding minimal scenarios that result in system failure or minimal scenarios ensuring that the system will be working. In case of MSSs, this task can be generalized as identification of minimal scenarios whose occurrence causes that the system performance decreases below a given level or minimal scenarios ensuring that the system achieves its mission. Special instances of these scenarios are Minimal Cut Vectors (MCVs) and Minimal Path Vectors (MPVs). A MCV represents a situation in which an improvement (in case of MSSs) or repair (in case of BSSs) of any non-perfectly functioning component results in improvement/repair of the system. Similarly, a MPV agrees with a situation in which a degradation (in case of MSSs) or failure (in case of BSSs) of any working component causes system degradation/failure. Knowledge of the MCVs and MPVs can be used to estimate system availability [8 and 9] or to find components with the greatest influence on the system [11 and 12]. However, the main issue is their detection.

---

\* Miroslav Kvassay, Jozef Kostolny

Department of Informatics, Faculty of Management Science and Informatics, University of Zilina, Slovakia  
E-mail: Miroslav.Kvassay@fri.uniza.sk

There exist a lot of algorithms for finding the MCVs and MPVs or their equivalents known as minimal cut sets and minimal path sets in case of BSSs [13 - 18]. Several algorithms have also been proposed for identification of the MCVs or MPVs in a MSS [19 - 21]. However, these algorithms have been developed specially for network systems and, therefore, there can be some complications to apply them to other types of MSSs. Because of that, new methods for identification of the MCVs and MPVs of a MSS have been considered in [12, 22 and 23]. These methods are based on the investigation of system structure using logical differential calculus [24]. However, time complexity of these algorithms has not been studied in [12, 22 and 23] and, therefore, we decided to perform some experiments whose results are presented in this paper.

## 2. Mathematical Background

Reliability analysis of a system requires creation of its model. Every system consists of one or more components (elements that are assumed to be indivisible into smaller parts). This implies that one of the principal information that has to be carried by the model is dependency between system state and states of its components. This dependency is usually expressed using a special mapping that is known as structure function. For a general MSS consisting of  $n$  components, this function has the following form [9 and 12]:

$$\phi(x) = \phi(x_1, x_2, \dots, x_n) : \{0, 1, \dots, m_1 - 1\} \times \dots \times \{0, 1, \dots, m_n - 1\} \rightarrow \{0, 1, \dots, m - 1\}, \quad (1)$$

where  $m$  is a number of possible states of the system (state  $m-1$  corresponds to perfectly functioning while state 0 agrees with the complete failure of the system),  $m_i$  for  $i = 1, 2, \dots, n$ , represents number of states of the  $i$ -th system component (state  $m_i-1$  means that component  $i$  is perfectly functioning while state 0 means that it completely failed),  $x_i$  is a variable denoting state of the  $i$ -th system component, and  $x = (x_1, x_2, \dots, x_n)$  is a vector of components states (state vector). Specially, if all system components have the same number of states as the system, i.e.  $m_1 = m_2 = \dots = m_n = m$ , then the system is recognized as a homogeneous [9 and 10]. Using this notation, a BSS, which allows defining only two states in the system/components performance, can be viewed as a homogeneous system in which  $m = 2$ .

Two classes of systems can be recognized depending on the properties of the structure function - coherent and noncoherent. The coherent systems are characterized by the fact that their structure function is monotone [8 - 10], which means that there exist no circumstances under which a failure/degradation (repair/improvement) of any system component can cause system repair/improvement (failure/degradation). Although real examples of noncoherent systems exist, e.g. logic networks studied in [4], most

of the systems that are investigated by reliability engineers are coherent. Furthermore, MCVs and MPVs, which are studied in this paper, have been defined in terms of coherency and, therefore, only coherent systems will be considered in what follows.

The structure function defines system topology. Its analysis can reveal conditions under which a failure (repair) of a binary-state component or degradation (improvement) of a multi-state component results in a failure (repair) of the BSS or in degradation (improvement) of the MSS. Because of that, it is useful in the qualitative analysis of BSSs and MSSs. However, only its knowledge is not sufficient for quantitative analysis for which the probabilities of individual states of individual system components have to be known:

$$p_{i,s} = \Pr\{x_i = s\}, \sum_{s=0}^{m_i-1} p_i = 1, \quad (2)$$

$$i = 1, 2, \dots, n, s = 0, 1, \dots, m_i - 1.$$

The knowledge of these probabilities allows us to calculate the system availability (unavailability) defined as the probability that the system can (cannot) satisfy a demand requiring at least level  $j$  of system performance [9 and 10], or to analyze importance of individual system components or their states for system activity. This investigation is done using some special measures, such as Birnbaum's, criticality, or Fussell-Vesely's importance [11, 12, 25 and 26]. Computation of some of these measures, e.g. the Fussell-Vesely's importance, requires knowledge of MCVs or MPVs [12].

### 2.1. Minimal Cut Vectors and Minimal Path Vectors

A MCV for system state  $j$  (for  $j = 1, 2, \dots, m-1$ ) is defined as a state vector  $x$  such that  $\phi(x) < j$  and  $\phi(y) \geq j$  for any  $y > x$  where notation  $y > x$  between state vectors  $x = (x_1, x_2, \dots, x_n)$  and  $y = (y_1, y_2, \dots, y_n)$  expresses that  $y_i \geq x_i$  for  $i = 1, 2, \dots, n$  and there exists at least one  $i$  such that  $y_i > x_i$  [9 and 10]. This definition implies that a MCV for a given system state corresponds to such situation in which a minor improvement (i.e. improvement by one state) of any non-perfectly working component causes that the system achieves at least state  $j$ . (Please note that component  $i$  is non-perfectly working if it is in a state that is less than  $m_i-1$ .)

Similarly, a MPV for state  $j$  (for  $j = 1, 2, \dots, m-1$ ) of a MSS is defined as a state vector  $x$  such that  $\phi(x) \geq j$  and  $\phi(y) < j$  for any  $y < x$  [9 and 10]. Using this definition, it can be shown easily that a MPV for a given system state represents such situation in which a minor degradation (i.e. degradation by one state) of any working component (i.e. component that is not in state 0) causes that the system state decreases below value  $j$ .

## 2.2. Logical Differential Calculus and Integrated Direct Partial Logic Derivatives

Logical differential calculus has been developed for analysis of dynamic properties of Boolean and Multiple-Valued Logic (MVL) functions. Direct Partial Logic Derivatives (DPLDs) are a special part of this tool. They allow finding situations in which a change of a value of a Boolean (MVL) variable results in a change of the analyzed Boolean (MVL) function [24].

The use of DPLDs in reliability analysis of BSSs and MSSs has been considered in several works. In [25], DPLDs have been used in the analysis of BSSs to detect situations in which a failure of a component results in system failure. In [26], the use of DPLDs to investigate circumstances under which a degradation of a multi-state component causes degradation of a MSS has been considered. These two works have primarily focused on homogeneous systems because the structure function of such systems can be recognized as a Boolean function (in the case of BSSs) or as a MVL function (in the case of MSSs). Other works, such as [12, 23 and 27], have considered application of DPLDs in the analysis of non-homogeneous systems. Based on these works, a DPLD investigating the structure function of a non-homogeneous system should be defined in the following way:

$$\frac{\partial \phi(j \rightarrow h)}{\partial x_i(s \rightarrow r)} = \begin{cases} 1 & \text{if } \phi(s_i, \mathbf{x}) = j \text{ and } \phi(r_i, \mathbf{x}) = h, \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

$s, r \in \{0, 1, \dots, m_i - 1\}, s \neq r; j, h \in \{0, 1, \dots, m - 1\}, j \neq h.$

Clearly, the nonzero elements of this derivative identify situations in which change of component  $i$  from state  $s$  to  $r$  causes change of the system state from value  $j$  to  $h$ . Since only coherent systems are studied in this paper, only DPLDs in which  $j > h$  and  $s > r$  or DPLDs in which  $j < h$  and  $s < r$  can be nonzero. The former can be used to find correlation between degradation of component  $i$  and system degradation while the latter can reveal circumstances under which the improvement of component  $i$  results in system improvement. This fact and the interpretation of MCVs and MPVs indicate that DPLDs  $\partial \phi(j \rightarrow h) / \partial x_i(s \rightarrow s + 1)$  where  $j < h$  could be used to detect the MCVs and DPLDs  $\partial \phi(j \rightarrow h) / \partial x_i(s \rightarrow s - 1)$  in which  $j > h$  to find the MPVs. This idea has been considered for homogeneous MSSs in [22] and for non-homogeneous systems in [12 and 23]. In these papers, it has been shown that direct use of DPLDs to identify MCVs or MPVs can be quite complicated and, therefore, new types of DPLDs have been introduced:

$$\frac{\partial_e \phi(h_{<j} \rightarrow h_{\geq j})}{\partial_e x_i(s \rightarrow r)} = \begin{cases} 1 & \text{if } x_i = s \text{ and } \phi(s_i, \mathbf{x}) < j \text{ and } \phi(r_i, \mathbf{x}) \geq j, \\ 0 & \text{if } x_i = s \text{ and } (\phi(s_i, \mathbf{x}) \geq j \text{ or } \phi(r_i, \mathbf{x}) < j), \\ * & \text{if } x_i \neq s, \end{cases} \quad (4)$$

$$\frac{\partial_e \phi(h_{\geq j} \rightarrow h_{<j})}{\partial_e x_i(s \rightarrow r)} = \begin{cases} 1 & \text{if } x_i = s \text{ and } \phi(s_i, \mathbf{x}) \geq j \text{ and } \phi(r_i, \mathbf{x}) < j, \\ 0 & \text{if } x_i = s \text{ and } (\phi(s_i, \mathbf{x}) < j \text{ or } \phi(r_i, \mathbf{x}) \geq j), \\ * & \text{if } x_i \neq s, \end{cases} \quad (5)$$

$s, r \in \{0, 1, \dots, m_i - 1\}, s \neq r; j \in \{1, 2, \dots, m - 1\}$ , where notation  $h_{<j}$  expresses that all system states that are less than  $j$  are investigated while notation  $h_{\geq j}$  implies that we take into account all situations in which the structure function has at least value  $j$ . These derivatives will be referred to as expanded Integrated Direct Partial Logic Derivatives (IDPLDs) because (a) they expand classical DPLDs using value “\*”, and (b) it can be shown that they combine (integrate) several DPLDs together.

## 3. Algorithms for Identification of Minimal Cut Vectors and Minimal Path Vectors based on Integrated Direct Partial Logic Derivatives

The definition of expanded IDPLD (4) introduced in the previous section implies that value “1” of this IDPLD detects situations in which change of the  $i$ -th system component from state  $s$  to  $r$  causes that the system achieves at least state  $j$ , value “0” agrees with situations in which the considered change of component state does not result in achieving at least system state  $j$ , and value “\*” implies that the component cannot change from state  $s$  to  $r$  because it is not in state  $s$ . It has been shown in [12 and 22] that if we compute expanded IDPLDs of the form of  $\partial_e \phi(h_{<j} \rightarrow h_{\geq j}) / \partial_e x_i(s \rightarrow s + 1)$  for all possible values of  $i$  and  $s$ , i.e. for  $i = 1, 2, \dots, n$  and for  $s = 0, 1, \dots, m_i - 2$ , and then calculate some form of their conjunction, i.e.  $\Pi$ -conjunction defined in Table 1-a, then we can identify all situations in which a minor improvement of any non-perfectly working component causes that the system achieves at least state  $j$ . This agrees with the interpretation of a MCV for system state  $j$  and, therefore, the MCVs for system state  $j$  can be computed using the algorithm presented in Fig. 1. If we repeat this algorithm for all possible system states, i.e. for  $j = 1, 2, \dots, m - 1$ , then all MCVs of the MSS can be found. Using IDPLDs (5) and  $\Pi$ -conjunction shown in Table 1-b, the similar algorithm can be derived for identification of the MPVs (Fig. 2).

## 4. Experimental Investigation of Algorithms Performance

The aforementioned algorithms for identification of the MCVs and MPVs are very similar. In fact, the only difference is that the algorithm for finding the MCVs uses expanded IDPLDs of the form of  $\partial_e \phi(h_{<j} \rightarrow h_{\geq j}) / \partial_e x_i(s \rightarrow s + 1)$ , while the algorithm for computation of the MPVs uses expanded IDPLDs

Π-conjunction of two expanded IDPLDs

Table 1

Table 1-a		$\frac{\partial_e \phi(h_{<j} \rightarrow h_{\geq j})}{\partial_e x_k(s_k \rightarrow r_k)}$		
		*	0	1
$\frac{\partial_e \phi(h_{<j} \rightarrow h_{\geq j})}{\partial_e x_i(s_i \rightarrow r_i)}$	*	*	0	1
	0	0	0	0
	1	1	0	1

Table 1-b		$\frac{\partial_e \phi(h_{\geq j} \rightarrow h_{<j})}{\partial_e x_k(s_k \rightarrow r_k)}$		
		*	0	1
$\frac{\partial_e \phi(h_{\geq j} \rightarrow h_{<j})}{\partial_e x_i(s_i \rightarrow r_i)}$	*	*	0	1
	0	0	0	0
	1	1	0	1

1. Repeat the next two steps for all system components:
  - 1.1. Compute expanded IDPLDs  $\partial_e \phi(h_{<j} \rightarrow h_{\geq j}) / \partial_e x_i(s \rightarrow s + 1)$  for  $s = 0, 1, \dots, m_i - 2$ .
  - 1.2. Calculate Π-conjunction  $\prod_{s=0}^{m_i-2} \partial_e \phi(h_{<j} \rightarrow h_{\geq j}) / \partial_e x_i(s \rightarrow s + 1)$  of the expanded IDPLDs computed in the previous step.
2. Calculate Π-conjunction of the Π-conjunctions computed in step 1 and identify state vectors for which it has value 1. These vectors correspond to the MCVs for system state  $j$ .

Fig. 1 Algorithm for finding all MCVs for state  $j$  of a MSS

1. Repeat the next two steps for all system components:
  - 1.1. Compute expanded IDPLDs  $\partial_e \phi(h_{\geq j} \rightarrow h_{<j}) / \partial_e x_i(s \rightarrow s - 1)$  for  $s = 1, 2, \dots, m_i - 1$ .
  - 1.2. Calculate Π-conjunction  $\prod_{s=1}^{m_i-1} \partial_e \phi(h_{\geq j} \rightarrow h_{<j}) / \partial_e x_i(s \rightarrow s - 1)$  of the expanded IDPLDs computed in the previous step.
2. Calculate Π-conjunction of the Π-conjunctions computed in step 1 and identify state vectors for which it takes value 1. These vectors correspond to the MPVs for system state  $j$ .

Fig. 2 Algorithm for finding all MPVs for state  $j$  of a MSS

of the form of  $\partial_e \phi(h_{\geq j} \rightarrow h_{<j}) / \partial_e x_i(s \rightarrow s - 1)$ . However, based on (4) and (5), there is no significant difference in their computation and, therefore, the algorithm for computation of the MPVs has the similar properties as the algorithm for finding the MCVs. Because of that, we investigated only the algorithm for identification of the MCVs. The study was done by performing some experiments on a computer with CPU Intel Core i5 (2.5 GHz), 4 GB RAM and Windows 7 OS. The algorithm was implemented in C++ programming language, and we analyzed its time complexity depending on the number of system components and depending on the number of all system MCVs. (For simplicity, we considered only homogeneous systems.) Firstly, the dependency of the computation time on the number of all system MCVs was studied. For such analysis, we generated systems with the same number of components but with the variable count

of the MCVs. Typical examples of such systems are  $k$ -out-of- $n$  homogeneous  $m$ -state systems that are defined as follows [28]:

- if at least  $k$  components are in state  $m - 1$ , then the system is in state  $m - 1$ ;
- else if at least  $k$  components are in state  $m - 2$  or better, then the system is in state  $m - 2$ ;
- ...
- else if at least  $k$  components are in state 1 or better, then the system is in state 1;
- else the system is in state 0.

The results of these experiments for 3- and 4-state homogeneous systems are presented in Figs. 3 and 4. Based on these figures, we can state that the running time of the algorithm does not depend on the number of the MCVs. On the other

hand, these figures indicate that the running time increases exponentially with the increasing number of system components. This is clearer in Fig. 5, in which the dependency of the running time on the number of system components for systems with the similar numbers of the MCVs is presented. This figure also implies that the algorithm depends exponentially on the number of components states and, therefore, it is not very appropriate for systems with a lot of components or systems containing components that have many states.

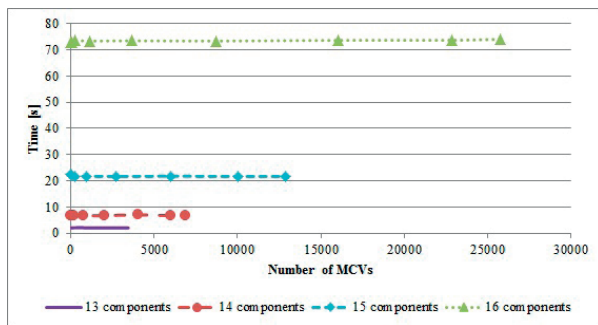


Fig. 3 Dependency of computation time on the number of the MCVs for homogeneous 3-state systems

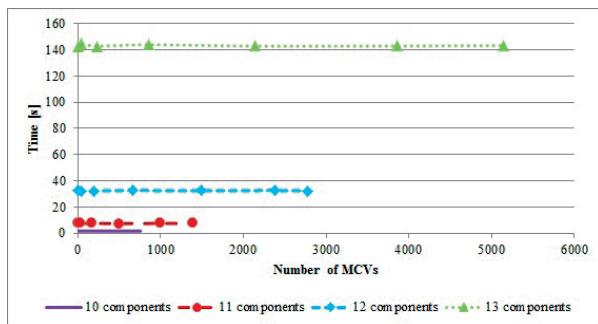


Fig. 4 Dependency of computation time on the number of the MCVs for homogeneous 4-state systems

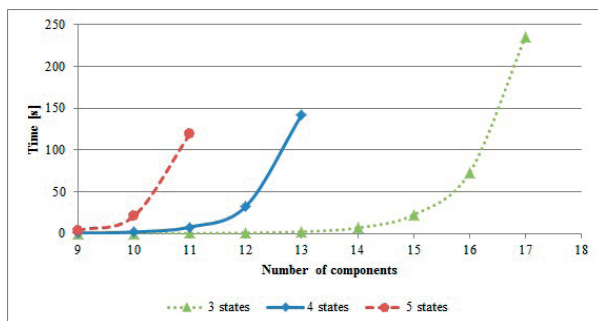


Fig. 5 Dependency of computation time on the number of system states for homogeneous multi-state systems with the similar numbers of the MCVs

## 5. Conclusion

MCVs and MPVs are one of the key concepts of reliability analysis. In the case of BSSs, a MCV represents such situation in which a repair of any failed component results in system repair, while a MPV corresponds to such situation in which a failure of any working component causes the system failure. In the case of MSSs, the MCVs for a given system state describe circumstances under which a minor improvement of any non-perfectly working component causes that the system achieves at least the considered state, while the MPVs for system state  $j$  agree with situations in which a minor degradation of any working component results in decrease in system state below value  $j$ .

MCVs and MPVs are very useful in reliability analysis. They can be used to estimate system availability or to analyze influence of individual system components on the system activity. Methods based on MCVs and MPVs could be used to analyze a large scale of systems, such as healthcare systems considered in [5 and 23], distribution networks presented in [6], or medical and temporal database systems studied in [29 and 30]. However, one of the main issues of use of these methods is identification of the MCVs and MPVs. For BSSs, many algorithms have been proposed for detection of the MCVs and MPVs or their equivalents known as minimal cut sets and minimal path sets. In the case of MSSs, some algorithms have also been designed. However, these algorithms are based on the assumption that the considered system is a network. Because of that, new algorithms that can be applied to a system of any type have been considered in [12, 22 and 23]. The mathematics behind these algorithms was summarized in this paper. Furthermore, some experiments estimating their time complexity were also done. The experiments showed that the running time of the algorithms does not depend on the number of system MCVs (MPVs) and, therefore, they are very suitable for the analysis of systems with a complicated structure, i.e. systems with a huge amount of the MCVs or MPVs. On the other hand, the experiments also showed that the running time depends exponentially on the number of system components and the number of components states and, therefore, the further research should focus on solving this issue.

## References

- [1] EISENBERG, N. A., SAGAR, B.: Importance Measures for Nuclear Waste Repositories, *Reliability Engineering & System Safety*, vol. 70, No. 3, pp. 217-239, 2000.
- [2] RADOS, I., SCHWARTZ, L.: The Worst Availability as a Parameter for Designing and Reporting on the Network Performances, *Communications - Scientific Letters of the University of Zilina*, vol. 13, No. 1, pp. 60-66, 2011.
- [3] BRIS, R.: Assessment of the Availability of an Offshore Installation by Stochastic Petri Net Modeling, *Communications - Scientific Letters of the University of Zilina*, vol. 16, No. 1, pp. 90-96, 2014.
- [4] ZAITSEVA, E., KVASSAY, M., LEVASHENKO, V., KOSTOLNY, J.: Reliability Analysis of Logic Network with Multiple Outputs, *Communications - Scientific Letters of the University of Zilina*, vol. 17, No. 1A, pp. 44-50, 2015.
- [5] ZAITSEVA, E., LEVASHENKO, V., RUSIN, M.: *Reliability Analysis of Healthcare System*, 2011 Federated Conference on Computer Science and Information Systems, FedCSIS 2011, 2011, pp. 169-175.
- [6] PRAKS, P., KOPUSTINSKAS, V.: *Monte-Carlo Based Reliability Modelling of a Gas Network Using Graph Theory Approach*, Ninth Intern. Conference on Availability, Reliability and Security, 2014, pp. 380-386.
- [7] ZIO, E.: Reliability Engineering: Old Problems and New Challenges, *Reliability Engineering & System Safety*, vol. 94, No. 2, pp. 125-141, 2009.
- [8] RAUSAND M., HOYLAND, A.: *System Reliability Theory*, 2<sup>nd</sup> ed., Hoboken: John Wiley & Sons, 2004.
- [9] LISNIANSKI, A., FRENKEL, I., DING, Y.: *Multi-state System Reliability Analysis and Optimization for Engineers and Industrial Managers*. London : Springer-Verlag, 2010.
- [10] NATVIG, B.: *Multistate Systems Reliability Theory with Applications*. Chichester : John Wiley & Sons 2011.
- [11] KUO, W., ZHU, X.: *Importance Measures in Reliability, Risk, and Optimization: Principles and Applications*. Chichester : Wiley, 2012.
- [12] KVASSAY, M., ZAITSEVA, E., LEVASHENKO, V.: Minimal Cut Sets and Direct Partial Logic Derivatives in Reliability Analysis, *Safety and Reliability: Methodology and Applications, Proc. of the European Safety and Reliability Conference, ESREL 2014*, 2015, pp. 241-248.
- [13] VATN, J.: Finding Minimal Cut Sets in a Fault Tree, *Reliability Engineering & System Safety*, vol. 36, No. 1, pp. 59-62, 1992.
- [14] RAUZY, A.: New Algorithms for Fault Trees Analysis, *Reliability Engineering & System Safety*, vol. 40, No. 3, pp. 203-211, 1993.
- [15] SINNAMON R. M., ANDREWS, J. D.: New Approaches to Evaluating Fault Trees, *Reliability Engineering & System Safety*, vol. 58, No. 2, pp. 89-96, 1997.
- [16] SHEN, Y.: A New Simple Algorithm for Enumerating All Minimal Paths and Cuts of a Graph, *Microelectronics Reliability*, vol. 35, No. 6, pp. 973-976, 1995.
- [17] YEH, W.-C.: Search for all MCs in Networks with Unreliable Nodes and Arcs, *Reliability Engineering & System Safety*, vol. 79, No. 1, pp. 95-101, 2003.
- [18] EMADI, A., AFRAKHTE, H.: A Novel and Fast Algorithm for Locating Minimal Cuts up to Second Order of Undirected Graphs with Multiple Sources and Sinks, *Intern. J. of Electrical Power & Energy Systems*, vol. 62, pp. 95-102, 2014.
- [19] YEH, W.-C.: A New Approach to the d-MC Problem, *Reliability Engineering & System Safety*, vol. 77, No. 2, pp. 201-206, 2002.
- [20] YEH, W.-C.: The Extension of Universal Generating Function Method to Search for All One-to-Many d-Minimal Paths of Acyclic Multi-State-Arc Flow-Conservation Networks, *IEEE Transactions on Reliability*, vol. 57, No. 1, pp. 94-102, 2008.
- [21] YEH, W.-C.: A Fast Algorithm for Searching All Multi-state Minimal Cuts, *IEEE Transactions on Reliability*, vol. 57, No. 4, pp. 581-588, 2008.
- [22] KVASSAY, M., ZAITSEVA, E., LEVASHENKO, V., KOSTOLNY, J.: *Minimal Cut Vectors and Logical Differential Calculus*, 2014 IEEE 44th Intern. Symposium on Multiple-Valued Logic, 2014, pp. 167-172.
- [23] KVASSAY, M., ZAITSEVA, E.: *Construction of Healthcare System Structure for Reliability Analysis*, 2014 Federated Conference on Computer Science and Information Systems, FedCSIS 2014, 2014, vol. 2, pp. 191-199.
- [24] YANUSHKEVICH, S. N., MILLER, D. M., SHMERKO, V. P., STANKOVIC, R. S.: *Decision Diagram Techniques for Micro- and Nanoelectronic Design Handbook*, vol. 2, Boca Raton: CRC Press, 2005.
- [25] ZAITSEVA, E. N., LEVASHENKO, V. G.: Importance Analysis by Logical Differential Calculus, *Automation and Remote Control*, vol. 74, No. 2, pp. 171-182, 2013.
- [26] ZAITSEVA, E., LEVASHENKO, V.: Multiple-valued Logic Mathematical Approaches for Multi-state System Reliability Analysis, *J. of Applied Logic*, vol. 11, No. 3, pp. 350-362, 2013.
- [27] ZAITSEVA, E., LEVASHENKO, V.: Multi-state System Analysis based on Multiple-valued Decision Diagram, *J. of Reliability and Statistical Studies*, vol. 5, No. Special, pp. 107-118, 2012.

- [28] BOEDIGHEIMER, R. A., KAPUR, K. C.: Customer-driven Reliability Models for Multistate Coherent Systems, *IEEE Transactions on Reliability*, vol. 43, No. 1, pp. 46-50, 1994.
- [29] KVET, M., VAJSOVA, M., MATIASKO, K.: Complex Data Management in MRI Results Processing, in *Applications of Computational Intelligence in Biomedical Technology*, BRIS, R., MAJERNIK, J., PANCERZ, K., ZAITSEVA, E., Eds. Springer International Publishing, 2016, pp. 119-141.
- [30] KVET, M., MATIASKO, K.: Column Level Uni-temporal Data, *Communications - Scientific Letters of the University of Zilina*, vol. 16, No. 1, pp. 97-104, 2014.