Reliability Analysis based on Logical Differential Calculus and Minimal Cut Set Methods

Miroslav Kvaššay *
Department of informatics
Faculty of Management Science and Informatics
University of Žilina in Žilina
Univerzitná 8215/1, 010 26 Žilina, Slovakia
miroslav.kvassay@fri.uniza.sk

Abstract
This paper focuses on qualitative and quantitative analysis of influence of system components on system reliability/availability. This analysis is implemented based on the concept of minimal cut sets (minimal cut vectors) and minimal path sets (minimal path vectors). New mathematical method is proposed for calculation of minimal cut vectors and minimal path vectors. This method is based on direct partial logic derivatives, which are part of logical differential calculus. The principal goals of this work are development of new algorithms for calculation of minimal cut vectors and minimal path vectors for any system that can be represented as a binary- or multi-state system and use them in computation of measures for estimation of system reliability/availability. These goals result in solving the following three problems: detailed analysis of influence of system components on system reliability/availability, development of universal mathematical background for calculation of minimal cut vectors and minimal path vectors for binary- and multi-state systems; and creating new algorithms for calculation of measures for system reliability/availability estimation.

Categories and Subject Descriptors
C.4 [Performance of Systems]: Reliability, availability, and serviceability; G.3 [Probability and Statistics]: Reliability and life testing

Keywords
reliability analysis, binary-state system, multi-state system, minimal cut set, minimal path set, logical differential calculus, importance measures

1. Introduction

The principal step in reliability analysis is construction of mathematical model. As a rule, two models are preferred in reliability engineering: Binary-State Systems (BSSs) and Multi-State Systems (MSSs). A BSS can be in one of two possible states – functioning and failed. The main problem of BSSs is to define the boundary between these two states, i.e., circumstances under which the system can be considered functioning and when it is failed. Therefore, BSSs are useful in identification and analysis of situations that result in the total failure of the system, or that cause deviation of the system from its perfect functioning [16, 19, 32]. A MSS allows defining more than two states in system performance and, therefore, it can be used to analyze processes that cause gradual degradation of the system [16, 17, 32].

Minimal Cut Sets (MCSs) and Minimal Path Sets (MPVs) are one of the principal concepts of reliability engineering. They have been used for both mathematical models, but most papers related to MCSs and MPVs consider their application in qualitative and quantitative analysis of BSSs, e.g., [2, 3, 5, 19, 20, 25]. In these cases, MCSs represent minimal sets of system components whose simultaneous failure results in system failure while MPVs correspond to minimal sets of components whose simultaneous work ensures that the system is functioning. This indicates that MCSs and MPVs can be used to evaluate system reliability from different points of view, e.g., they allow estimating system availability/reliability, or they can be used to identify components that have the most influence on system activity. Therefore, they are very useful in reliability analysis of BSSs.

The concept of MCSs and MPVs is used also in reliability analysis of MSSs. In this case, MCSs and MPVs, or their alternatives known as Minimal Cut Vectors (MCVs) and Minimal Path Vectors (MPVs) respectively, are primarily used in reliability evaluation of network systems, e.g., [4, 23, 24, 26]. This indicates that the existing algorithms for identification of MCVs (MPVs) are designed for systems that can be presented as a network and, therefore, they cannot be used for application of MCVs and MPVs methods to other types of MSSs. Because of that, development of algorithms for calculation of MCVs (MPVs) for systems of any type is an actual problem in reliability analysis. Different mathematical methods can be used to solve this task. One of them is logical differential calculus.

Logical differential calculus has originally been developed for analysis of dynamic properties of Boolean and multiple-
2. Reliability Analysis and Logical Differential Calculus

Two different types of models are used in reliability analysis (Figure 2): Binary-State Systems (BSSs) and Multi-State Systems (MSSs). A BSS allows defining only two states in system/component (a basic part of the system that is assumed to be indivisible into smaller parts) performance: functioning (state 1) and failed (state 0). The correlation between system state and states of system components is defined by the structure function:

\[ \phi(x) = \phi(x_1, x_2, \ldots, x_n) : \{0, 1\}^n \rightarrow \{0, 1\}, \] (1)

where \( n \) is a number of system components, \( x_i \) for \( i = 1, 2, \ldots, n \) is a state of the \( i \)-th component, and \( x = (x_1, x_2, \ldots, x_n) \) is a vector of system states (state vector). An example of a BSS is a system of two hard disk drives (components) combined into RAID 1 configuration, which is working if at least one disk is working (Figure 3).

Please note that the structure function of a BSS can be interpreted as a Boolean function, which allows us to use some concepts of Boolean functions in reliability analysis of BSSs.

A MSS makes possible defining more than only two states in system/component behavior. Therefore, its structure function has the following form:

\[ \phi(x) = \phi(x_1, x_2, \ldots, x_n) : \{0, 1, \ldots, m_1 - 1\} \times \{0, 1, \ldots, m_2 - 1\} \times \cdots \rightarrow \{0, \ldots, m - 1\}. \] (2)

where \( n \) is a number of system components, \( x_i \) for \( i = 1, 2, \ldots, n \) is a state of component \( i \), \( x = (x_1, x_2, \ldots, x_n) \) is a vector of system states (state vector), \( m_i \) defines number of states of component \( i \) (state \( m_i - 1 \) – perfectly functioning, state 0 – completely failed) and \( m \) is a number of system states (state \( m - 1 \) – perfectly functioning, state 0 – completely failed). As an example, consider a simple parallel system in Figure 4 consisting of two components. This system represents a network that is used to transport some commodity from point A to point B; therefore, its performance depends on the number of working lines (system components). According to this, the system structure function has the form defined by table in Figure 4.

Homogeneous MSSs are a special class of MSSs. A homogeneous MSS contains only components that have the same number of states as the system, i.e., \( m_1 = m_2 = \cdots = m_n = m \). The structure function of a homogeneous MSS is very similar to the definition of a Multiple-Valued Logic (MVL) function, which allows us to use some techniques of MVL in reliability analysis of MSSs.

Figure 1: Research topic

Figure 2: System models

Figure 3: A parallel system modeled as a BSS

Figure 4: A parallel system modeled as a MSS

Table 1: truth table of function \( \phi(x_1, x_2) \)

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( \phi(x_1, x_2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
Reliability analysis primarily focuses on coherent systems. A coherent system is defined as a system whose structure function is monotonic. This implies that a failure (degradation) of any system components cannot result in system failure (degradation). If this assumption is not satisfied, then the system is recognized as noncoherent. Noncoherent systems are not very common in reliability analysis and, therefore, in what follows, we will assume that the systems are coherent.

2.1 Logical Differential Calculus

Logical differential calculus is a tool that has been developed for analysis of dynamic properties of logic functions. The central term of this tool is a derivative that is defined for Boolean function \( f(x) \) as follows [22]:

\[
\frac{\partial f(x)}{\partial x_i} = f(1, x_i) \oplus f(0, x_i),
\]

(3)

where \( \oplus \) is a symbol of logic operation XOR and \( f(s, \mathbf{x}) = (x_1, x_2, \ldots, x_{i-1}, s, x_{i+1}, \ldots, x_n) \) for \( s \in \{0, 1\} \).

The definition of the Boolean derivative implies that it identifies situations in which a change of the value of Boolean variable \( x_i \) results in a change of the value of Boolean function \( f(x) \). However, it does not allow identifying whether the Boolean function changes from value 1 to 0 or vice versa. Because of that, another type of Boolean derivatives has been introduced. These derivatives are known as Direct Partial Boolean Derivatives (DPBDs), and they are defined as follows [22]:

\[
\frac{\partial f(j \rightarrow \overline{j})}{\partial x_i(s \rightarrow \overline{s})} = \begin{cases} 
1 & \text{if } f(s, \mathbf{x}) = j \text{ and } f(s, \mathbf{x}) = \overline{j} \\
0 & \text{else}
\end{cases}
\]

(4)

for \( s, j \in \{0, 1\} \).

Clearly, there exist four different DPBDs. In the case of BSSs with the structure function \( \phi(x) \), Boolean derivatives \( \partial \phi(1 \rightarrow 0)/\partial x_i(1 \rightarrow 0) \) and \( \partial \phi(0 \rightarrow 1)/\partial x_i(0 \rightarrow 1) \) can be used to find scenarios when a failure (repair) of component \( i \) causes that the system fails (will be repaired) (Figure 5). Similarly, DPBDs \( \partial \phi(0 \rightarrow 1)/\partial x_i(1 \rightarrow 0) \) and \( \partial \phi(1 \rightarrow 0)/\partial x_i(0 \rightarrow 1) \) identify situations in which component failure (repair) results in system repair (failure). However, these two derivatives are irrelevant in the analysis of coherent BSSs because only the first two derivatives can be nonzero for such systems [31].

Several types of logic derivatives exist in the case of MVL functions. For our purposes, Direct Partial Logic Derivatives are the most important. These derivatives are defined for MVL function \( f_{m}(\mathbf{x}) \) as follows [22]:

\[
\frac{\partial f_{m}(j \rightarrow h)}{\partial x_i(s \rightarrow r)} = \begin{cases} 
1 & \text{if } f_{m}(s, \mathbf{x}) = j \text{ and } f_{m}(r, \mathbf{x}) = h \\
0 & \text{else}
\end{cases}
\]

(5)

for \( s, r, h, j \in \{0, 1, \ldots, m - 1\} \), \( s \neq r, j \neq h \). Here, \( f_{m}(a, \mathbf{x}) = (x_1, x_2, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_n) \) for \( a \in \{0, 1\} \).

DPBDs and DPLDs are defined only in such points of Boolean or MVL function in which variable \( x_i \) takes value \( s \) since these derivatives assume that the variable changes from value \( s \) to another.

In the case of homogeneous systems, a DPLD can be used to find situations in which change of state of component \( i \) from value \( s \) to \( r \) results in the change of system state from \( j \) to \( h \) (Figure 6). It is clear that only DPLDs that meet the next assumptions can be nonzero in the case of coherent systems:

a) \( s > r \) and \( j > h \) (identification of situations in which component degradation results in system degradation);

b) \( r > s \) and \( h > j \) (identification of situations in which component improvement results in system improvement).

2.2 Qualitative Analysis of Binary-State Systems

The qualitative analysis focuses on identification of situations that causes system failure or repair of its activity. Its basic terms are Minimal Cut Sets (MCSs), Minimal Path Sets (MPSs), Minimal Cut Vectors (MCVs), Minimal Path Vectors (MPVs) and component criticality [19].

A cut set is a set of components whose simultaneous failure results in system failure. If no component can be removed from a cut set without losing its status as a cut set, then the cut set is minimal. Similarly, a (minimal) path set corresponds to a (minimal) set of components whose simultaneous functioning ensures that the system is working.

A MCV agrees with a situation in which a repair of any failed component results in system repair. A MPV repre-
Component is critical when its failure (repair) causes that the system fails (will be repaired). If component \( i \) is critical at state vector \((x_1, x_2, \ldots, x_i-1, x_{i+1}, \ldots, x_n)\), then state vector \((1, x)\) is recognized as critical path vector for component \( i \) and state vector \((0, x)\) is known as critical cut vector for the component.

The relation between DPBDs and critical path (cut) vectors has been studied in work [31]. It has been shown that nonzero values of DPBD \( \partial \phi (1 \to 0) / \partial x_i (1 \to 0) \) correspond to critical path vectors for component \( i \) while nonzero values of DPBD \( \partial \phi (0 \to 1) / \partial x_i (0 \to 1) \) coincide with critical cut vectors. Therefore, DPBDs can be used in qualitative analysis of BSSs based on the concept of component criticality. However, the considered paper has not studied the relationship between MCVs (MPVs) and DPBDs. Since MCVs and MPVs are very important in reliability analysis, the first goal of our research was to find whether DPBDs can be used to identify them. This result could be used not only in qualitative analysis of BSSs but also in their quantitative analysis.

### 2.3 Quantitative Analysis of Binary-State Systems

The quantitative analysis allows studying system from the point of view of probability theory. For this purposes, the probabilities of component states have to be known:

\[ p_i = \Pr(x_i = 1), \quad p_0 = \Pr(q_i = 0), \quad p_i + q_i = 1. \]  

Please note that the probability \( p_i \) is known as availability of component \( i \) while \( q_i \) is identified as its unavailability.

The knowledge of system structure and availabilities of its components allows us to compute system availability and unavailability:

\[ A(p) = \Pr\{\phi(x) = 1\}, \quad U(q) = \Pr\{\phi(x) = 0\}, \quad A(p) + U(q) = 1, \]  

where \( p = (p_1, p_2, \ldots, p_n) \) is a vector of probabilities of system components and \( q = (q_1, q_2, \ldots, q_n) \) is a vector of their unavailabilities.

System availability is one of the basic measures used in reliability analysis. It can also be used to compute other measures, such as, mean time to system failure or mean time to system repair. However, it does not permit quantifying influence of system components on system activity.

To this purpose, other measures are used. These measures are known as Importance Measures (IMs) [10].

IMs are used to quantify the correlation between component failure (repair) and system failure (repair). The most commonly known IMs are Structural Importance (SI), Birnbaum’s Importance (BI), Criticality Importance (CI), and Fussell-Vesely’s Importance (FVI). Their definitions and meaning are presented in Table 1. The IMs can be divided into two groups. The first group is composed of IMs based on critical path (cut) vectors (SI, BI, CI, etc.). These IMs analyze from different points of view the probability that a given component is critical for system failure (functioning). The second group contains IMs that quantify contribution of a given component to system failure (functioning). The typical example of such IMs is the FVI. (Please note that a component contributes to system failure (functioning) if at least one MCS (MPS) containing the component is failed (functioning) [10] (expression MCs(i) in the definition of the FVI).)

Since the definitions of the SI and BI correspond to quantifying situations in which component \( i \) is critical, they can also be computed based on DPBDs (Table 2) [31]. Before this thesis, it was not known how to compute the FVI based on DPBDs since the relation between MCVs (MPVs) was not identified. Therefore, finding dependency between MCVs (MPVs) and DPBDs allows us to propose new formulae for computation of the FVI that will be based on logical differential calculus. And, this was another goal of the thesis.

### 2.4 Availability of Multi-State Systems

In the case of MSSs, system availability is defined as the probability that the system is in an acceptable state. The acceptable state is a state in which the system can perform required tasks [16, 17]. If we assume that the system is functioning if it is at least in state \( j \), then the availability and unavailability are defined as follows [16, 17]:

\[ A^j(p) = \Pr\{\phi(x) \geq j\}, \quad U^{j+1}(q) = \Pr\{\phi(x) < j\}, \quad A^j(p) + U^{j+1}(q) = 1, \]  

where \( p = (p_1, p_2, \ldots, p_n) \) is a vector of probabilities of all states of all components of the MSS and vector \( p_i = (p_{i,0}, p_{i,1}, \ldots, p_{i,m_i-1}) \) for \( i \in \{1, 2, \ldots, n\} \) defines probabilities of all states of component \( i \), i.e., \( p_i = \Pr(x_i = s) \) for \( s = 0, 1, \ldots, m_i - 1 \). This definition indicates that it can be quite complicated to define the availability of a MSS without the knowledge of minimal requirements on system performance.

### 2.5 Qualitative Analysis of Multi-State Systems

Two different approaches exist in the qualitative analysis of MSSs. The first approach focuses on “system state” while the second is based on the definition of system availability. This can be seen in the definitions of component criticality (Table 3) when we distinguish between situations in which a component is critical for a given level of system availability [16] and in which it is critical for a concrete system state [21].

MCVs and MPVs are other terms of the qualitative analysis of MSSs [16]. Both, a MCV and MPV, are defined only with respect to system availability level. A MCV for level \( j \) of system availability agrees with a situation in which the system is in state below \( j \) and improvement of any non-perfectly working component causes that the system will be at least in state \( j \). Similarly, a MPV for level \( j \) of system availability corresponds to such situation in which the system is at least in state \( j \) and degradation of any component that can degrade results in decrease in system availability level below value \( j \). It follows that MCVs (MPVs) can be defined for \( j \in \{1, 2, \ldots, m - 1\} \). The concept of MCVs and MPVs also exists in reliability analysis of MSSs [1], but it is not used very often since definitions of MCVs and MPVs are quite complicated for MSSs and they have no deeper meaning for practice.

Use of DPLDs in the qualitative analysis of MSSs has been considered in several works, e.g., [27, 30]. These works focus on identification of critical path vectors for a system.
Table 1: Basic importance measures

<table>
<thead>
<tr>
<th>Importance Measure</th>
<th>Definition</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI</td>
<td>$SI_i = \sum_{(i;x)\in{0,1}^{n-1}}(\phi(1_i,x) - \phi(0_i,x)) / 2^{n-1}$</td>
<td>A relative number of situations in which a given component is critical for the system activity.</td>
</tr>
<tr>
<td>BI</td>
<td>$BI_i = Pr(\phi(1_i,x) - \phi(0_i,x) &gt; 0)$</td>
<td>The probability that the component is critical for the system.</td>
</tr>
<tr>
<td>CI</td>
<td>$CI_i = BI_i q_i / U(q)$</td>
<td>The probability that the system failure has been caused by the component failure given that the system has failed.</td>
</tr>
<tr>
<td>FVI</td>
<td>$FVI_i = Pr(MCSs(i)) / U(q)$</td>
<td>The probability that the component contributes to the system failure probability.</td>
</tr>
</tbody>
</table>

Table 2: Basic importance measures based on DPBDs

<table>
<thead>
<tr>
<th>Importance Measure</th>
<th>Definition</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI</td>
<td>$SI_i = TD(\frac{\partial\phi(1 \rightarrow 0)}{\partial x_i(1 \rightarrow 0)})$</td>
<td>TD(,)... the truth density of a given Boolean function, i.e. a relative number of situations in which the Boolean function is nonzero</td>
</tr>
<tr>
<td>BI</td>
<td>$BI_i = Pr(\frac{\partial\phi(1 \rightarrow 0)}{\partial x_i(1 \rightarrow 0)} \leftrightarrow 1)$</td>
<td>↔... symbol of logical biconditional</td>
</tr>
</tbody>
</table>

Table 3: Component criticality for MSSs

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path vector for level $j$ of system availability</td>
<td>$\phi(x) \geq j$</td>
<td>It corresponds to situations when the system is at least in state $j$.</td>
</tr>
<tr>
<td>Critical path vector for level $j$ of system availability and for state $s$ of component $i$</td>
<td>$\phi(s_i,x) \geq j$ and $\phi((s - 1)_i,x) &lt; j$</td>
<td>It corresponds to situations in which degradation of state $s$ of component $i$ results in degradation of system state below value $j$.</td>
</tr>
<tr>
<td>Cut vector for level $j$ of system availability</td>
<td>$\phi(x) &lt; j$</td>
<td>It corresponds to situations when the system is below state $j$.</td>
</tr>
<tr>
<td>Critical cut vector for level $j$ of system availability and for state $s$ of component $i$</td>
<td>$\phi(s_i,x) &lt; j$ and $\phi((s + 1)_i,x) \geq j$</td>
<td>It corresponds to situations in which improvement of state $s$ of component $i$ results in improvement of system state at least to value $j$.</td>
</tr>
<tr>
<td>Path vector for system state $j$</td>
<td>$\phi(x) = j$</td>
<td>It corresponds to situations when the system is in state $j$.</td>
</tr>
<tr>
<td>Critical path vector for system state $j$ and for state $s$ of component $i$</td>
<td>$\phi(s_i,x) = j$ and $\phi((s - 1)_i,x) &lt; j$</td>
<td>It corresponds to situations in which degradation of state $s$ of component $i$ results in degradation of system state $j$.</td>
</tr>
</tbody>
</table>
state in homogeneous MSSs in which a degradation of any system component by one state can cause system degradation by only one state. This implies that only DPLDs can be nonzero in such systems. However, these DPLDs are not sufficient for analysis of other types of systems. This implied another goal of our work that was proposing extension of DPLDs in such a way that they would be able to identify any type of critical state vectors regardless of type of the investigated system. Another point was to find the relation between DPLDs and MCVs (MPVs). Solutions to these two works allow us to propose a complex framework for qualitative analysis of MSSs that is based on one mathematical tool and that can be used to propose new algorithms for quantitative analysis of MSSs.

2.6 Importance Analysis of Multi-State Systems
In the case of MSSs, it is also important to find components that have the most influence on system performance. There exist a lot of generalizations of IMs from BSSs on MSSs, e.g., [1, 6, 21, 30]. In these works, several approaches of how to extend IMs on MSSs can be recognized. Firstly, we can analyze influence of a given component state on a given system state (availability level) [1, 30]. Another possibility is to identify the importance of a given component state on the entire system (not only on a concrete system state) or to find total influence of a given component on a concrete system state (availability level) [6]. Finally, we can also analyze the total importance of a given component on the whole system [21]. Therefore, it would be useful to propose a universal method that could be used to analyze all possible dependencies. Logical differential calculus can be a good choice for this task. Its use in importance analysis of MSS has been considered, for example, in work [30]. In this work, computation of the SI and BI using DPLDs has been proposed for the special type of homogeneous systems. Our goal was to extend this method on a MSS of any type. Also, another objective was to propose the FVI computation based on logical differential calculus since this IM has not been considered in the aforementioned work.

Relations between MCVs (MPVs) and DPBDs have been studied in works [11, 28]. Firstly, recall that a MCV corresponds to a situation in which a repair of any failed component results in system repair. Since the influence of component i on system state can be modeled using DPBD ∂φ(0 → 1)/∂x_i(0 → 1), a state vector can be a MCV if and only if it contains at least one failed component and DPBD ∂φ(0 → 1)/∂x_i(0 → 1) computed with respect to any failed component has nonzero value for the considered state vector. This implies that a MCV can be identified as a point of the structure function at which every derivative ∂φ(0 → 1)/∂x_i(0 → 1) that can be computed in it (i.e., DPBDs with respect to every component that is in state 0) takes nonzero value (Figure 7). For this purpose, we have proposed expanded DPBDs in paper [28] that are defined for Boolean function f(x) as follows:

\[
\frac{\partial f(j \rightarrow \overline{j})}{\partial x_i(s \rightarrow \overline{s})} = \begin{cases} 
1 & \text{if } x_i = s \text{ and } f(s_i, x) = j \text{ and } f(\overline{s_i}, x) = \overline{j} \\
0 & \text{if } x_i = s \text{ and } (f(s_i, x) \neq j \text{ or } f(\overline{s_i}, x) \neq \overline{j}) \\
* & \text{if } x_i = \overline{s}
\end{cases}
\]  

(9)

Next, define a special conjunction of two expanded derivatives that will be referred to as \(\bigwedge\)-conjunction and whose computation for a coherent BSS with the structure function φ(x) is defined by Table 4. This conjunction has three possible values. In terms of reliability analysis, value 1 agrees with situations in which at least one component from \(i_1\) and \(i_2\) is in state 1 and its change to state \(\overline{1}\) causes the same change of the system state, e.g., for \(s = 0\), the value 1 corresponds to cut vectors at which at least one of components \(i_1\) and \(i_2\) can be critical, and all components from set \{{1, 2}\} that can be critical are critical. Value 0 corresponds to situations when at least one component can change from state 0 to \(\overline{1}\), but this change has no effect on system state. Finally, the value * agrees with state vectors in which no component from set \{{1, 2}\} can change into state \(\overline{1}\) because both are in this state. This implies that computation of \(\bigwedge\)-conjunction of all expanded DPBDs \(\bigwedge_e\phi(0 \rightarrow 1)/\bigwedge_e x_i(0 \rightarrow 1)\) allows us to identify situations in which a repair of any failed component results in system repair. This agrees with the meaning of MCVs and, therefore, we can state that the MCVs correspond to points at which expression \(i = 1 \bigwedge_e \phi(0 \rightarrow 1)/\bigwedge_e x_i(0 \rightarrow 1)\) takes value 1. For illustration, consider a series-parallel system depicted in Figure 8. The system is functioning if and only if component 1 is functioning and at least one of components 2 and 3 is functioning. This implies that the system MCVs are components sets \{{1, 2}, {2, 3}\}. These sets correspond to MCVs \({0, 1, 1}\) and \({1, 0, 0}\). If we want to find these MCVs based on logical differential calculus, then we have to compute expanded DPBDs \(\partial_e\phi(0 \rightarrow 1)/\partial_e x_i(0 \rightarrow 1)\) for \(i = 1, 2, 3\) and find their \(\bigwedge\)-conjunction. As we can see in Table 5, the \(\bigwedge\)-conjunction of these derivatives takes value 1 at points \({0, 1, 1}\) and \({1, 0, 0}\) of the structure function, therefore, these points are MCVs of the considered system. This result is in agreement with our expectations.

\[
\begin{array}{c|ccc}
\hline
\bigwedge\text{-conjunction} & \frac{\partial_e\phi(s_2 \rightarrow \overline{s}_2)}{\partial_e x_1(s_2 \rightarrow \overline{s}_2)} & \frac{\partial_e\phi(s_1 \rightarrow \overline{s}_1)}{\partial_e x_1(s_1 \rightarrow \overline{s}_1)} & \frac{\partial_e\phi(s_1 \rightarrow \overline{s}_1)}{\partial_e x_2(s_1 \rightarrow \overline{s}_1)} \\
\hline
* & 0 & 0 & 1 \\
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
1 & 0 & 0 & * \\
0 & 1 & 0 & * \\
1 & 0 & 1 & * \\
1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 \\
\hline
\end{array}
\]

Table 4: \(\bigwedge\)-conjunction of two expanded DPBDs

Figure 8: A series-parallel system

At the end of this section, please note that the MPVs of a BSS can be identified in the similar way. The only difference is that expanded DPBDs \(\partial_e\phi(1 \rightarrow 0)/\partial_e x_i(1 \rightarrow 0)\) have to be used because MPVs correspond to situations in which failure of any working component results in system failure.
3.1 Algorithm for Identification of Minimal Cut Vectors

The method for identification of MCVs described above can be implemented on a computer in the form of the following algorithm:

1. Compute expanded DPBDs
   \( \frac{\partial e \phi}{\partial e x_i} (0 \rightarrow 1) \) for all system components.
2. Using rules defined in Table 4, calculate \( \wedge \) conjunction of expanded DPBDs from the previous step, and identify state vectors for which it is 1. This state vectors correspond to MCVs of the system.

We analyzed time complexity of this algorithm by comparing it with another one from papers [7, 8]. The algorithm from the considered works is based on the problem of dualization of monotone Boolean functions which lies in identification of all prime implicants (implicates) of a monotone Boolean function if all prime implicants (implicates) of the dual function are known. (The dual function to a Boolean function \( f( x_1, x_2, \ldots, x_n) \) is a Boolean function of the form of \( f( \bar{x_1}, \bar{x_2}, \ldots, \bar{x_n}) \)). It can be shown simply that prime implicants correspond to MCSs while the meaning of prime implicates agree with the meaning of MPSs. This implies that this algorithm can also be used to find all system MCVs and MPVs.

We implemented both algorithms in C++ programing language, and the experiments were done on a computer with CPU Intel Core i5 2.5 GHz, 4 GB RAM and Windows 7 64 bit OS. The results of the experiments are presented in Figures 9 – 12. These figures analyze time complexity of the compared algorithms based on the number of MCVs that exist in a BSS and also based on the number of system components. The dependency between time complexity and the number of MCVs of these two algorithms is compared in Figures 9 and 10. It can be seen that the approach based on the expanded DPBDs (the solid line denoted as DPBD) does not depend on the number of MCVs while the algorithm based on the dualization of monotone Boolean functions [7, 8] (the dotted line denoted as MBFD) does. The graphs in Figures 11 and 12 present the dependency of computation time on the number of system components. In this situation, the compared systems have a different number of components, but they have the same counts of MCVs. As we can see, the algorithm [7, 8] for identification of MCVs (the dotted line denoted as MBFD) depends much more. So, the experiments have showed that our algorithm, which is based on logical differential calculus, is more appropriate for systems containing a huge amount of MCVs (MPVs) than the algorithm proposed in works [7, 8] and, therefore, it can be useful in reliability analysis of systems with complicated structure.

3.2 Fussell-Vesely Importance based on Logical Differential Calculus

It was showed in the dissertation thesis that the FVI can be computed using MCVs as follows:

\[
FVI_i = \frac{\Pr(\exists MCV(0_i) \in MCVs; \exists \leq MCV(0_i))}{U(q)},
\]

where \( U(q) \) is the reliability of the system.
where MCVs is a set of all MCVs, MCV(0, i) is a MCV in which $x_i = 0$, event $\{3MCV(0, i) \in MCVs; x \leq MCV(0, i)\}$ means that there is at least one MCV with $x_i = 0$ that is greater than or equal to an arbitrary state vector $x$ if and only if $x_i \leq y_i$ for every $i \in \{1, 2, \ldots, n\}$, and $U(q)$ denotes system unavailability. This definition of the FVI indicates that only MCVs that contain component $i$ in state 0 are needed to compute this IM. According to the definition of a MCV, DPBD $\partial f(0 \rightarrow 1)/\partial x_i(0 \rightarrow 1)$ has to be nonzero at points that correspond to MCVs with $x_i = 0$. This means that the MCVs needed for calculation of the FVI of component $i$ can be gained directly from DPBD $\partial f(0 \rightarrow 1)/\partial x_i(0 \rightarrow 1)$ without need to compute the expanded DPBDs and their $\cap$-conjunction. We showed in the dissertation thesis that the MCVs of the form of $(0, x)$ correspond to the elements of the following set:

$$\max \left\{ \arg \max_{x \in \{0, 1\}^n} \left( \frac{\partial f(0 \rightarrow 1)}{\partial x_i(0 \rightarrow 1)} \right) \right\}, \quad (11)$$

where $\arg \max()$ identifies all state vectors for which expression $\partial f(0 \rightarrow 1)/\partial x_i(0 \rightarrow 1)$ is nonzero while $\max()$ takes maximal from these state vectors in the sense of the relation "$<$" defined on the set of all state vectors. (This relation is defined for two state vector $x$ and $y$ as follows: $x < y$ if and only if $x_i \leq y_i$ for all $i \in \{1, 2, \ldots, n\}$ and there exists at least one $i$ such that $x_i < y_i$.) Therefore, it is possible to compute the FVI in the following way:

$$FV, I_i = \Pr \left\{ \exists y \in \max \left\{ \arg \max_{x \in \{0, 1\}^n} \left( \frac{\partial f(0 \rightarrow 1)}{\partial x_i(0 \rightarrow 1)} \right) ; x \leq y \right\} \right\} / U(q), \quad (12)$$

Finally, define the minimal negative function for Boolean function $f(x)$:

$$\text{MNF}(f(x)) = \begin{cases} 1 & \text{if } \exists y \in \left\{ \arg \max_{x \in \{0, 1\}^n} \left( \frac{\partial f(0 \rightarrow 1)}{\partial x_i(0 \rightarrow 1)} \right) \right\}; x \leq y \\ 0 & \text{else} \end{cases} \quad (13)$$

that represents a non-increasing Boolean function that can be created from the Boolean function $f(x)$ in such a way that we change value of the function $f(x)$ to value 1 in the minimal number of points at which the Boolean function takes value 0. Then, it can be shown that the FVI based on MCVs can be computed using only DPBDs and the concept of the minimal negative function in the following manner:

$$FV, I_i = \Pr \left\{ \text{MNF} \left( \frac{\partial f(0 \rightarrow 1)}{\partial x_i(0 \rightarrow 1)} \right) \right\} / U(q)$$

$$= q_i \Pr \left\{ \text{MNF} \left( \frac{\partial f(0 \rightarrow 1)}{\partial x_i(0 \rightarrow 1)} \right) \right\} / U(q), \quad (14)$$

where $q_i$ is unavailability of component $i$. So, we have obtained the new formula for calculation of the FVI that requires no a priori information about MCSs or MCVs. The similar relation can also be gotten for the FVI based on MPSs.

4. Multi-State Systems and Logical Differential Calculus

Use of DPLDs in reliability analysis of MSSs has been considered in papers [27, 30]. These works have consi-
dered only homogeneous MSSs in which a degradation of any system component by one state can result in degradation of system state at most by one state too. However, a lot of systems are non-homogeneous or do not satisfy this property.

4.1 Integrated Direct Partial Logic Derivatives and Critical State Vectors

It was showed in the thesis, that the following modification of the definition of DPLD allows us to use it in the analysis of non-homogeneous systems with the structure function $\phi(x)$:

$$ \frac{\partial \phi(j \rightarrow h)}{\partial x_i(s \rightarrow r)} = \begin{cases} 1 & \text{if } \phi(s, x) = j \text{ and } \phi(r, x) = h \\ 0 & \text{else} \end{cases} $$

for $s, r \in \{0, 1, \ldots, m_i - 1\}$ and $j, h \in \{0, 1, \ldots, m - 1\}, s \neq r, j \neq h$.

However, these derivatives are not very suitable for analysis of systems in which a degradation of component by one state (a minor degradation) can result in degradation of system state by more than one state. Therefore, a new type of DPLDs was defined in the dissertation thesis. These derivatives were named as Integrated Direct Partial Logic Derivatives (IDPLDs) since they combine several DPLDs together.

There exist three basic types of IDPLDs. IDPLDs of type I are suitable for analysis of consequences of component degradation on a concrete system state, and they are defined with respect to the structure function $\phi(x)$ as follows:

$$ \frac{\partial \phi(j \rightarrow h)}{\partial x_i(s \rightarrow r)} = \begin{cases} j^{-1} \frac{\partial \phi(j \rightarrow h)}{\partial x_i(s \rightarrow r)} & \text{if } h = 0 \\ 0 & \text{else} \end{cases} $$

$$ \frac{\partial \phi(j \rightarrow h)}{\partial x_i(s \rightarrow r)} = \begin{cases} 1 & \text{if } \phi(s, x) = j \text{ and } \phi(r, x) < j \\ 0 & \text{else} \end{cases} $$

$$ \frac{\partial \phi(j \rightarrow h)}{\partial x_i(s \rightarrow r)} = \begin{cases} m^{-1} \frac{\partial \phi(j \rightarrow h)}{\partial x_i(s \rightarrow r)} & \text{if } h = m_i - 1 \\ 0 & \text{else} \end{cases} $$

It is clear that the first derivative identifies situations in which change of the $i$-th component state from value $s$ to $r$ results in system degradation below state $j$ while the second derivative can be used to find situations in which the analyzed change of component $i$ causes that the system falls into state $j$.

IDPLDs of type II are defined with respect to the structure function $\phi(x)$ as follows:

$$ \frac{\partial \phi(j \rightarrow h)}{\partial x_i(s \rightarrow r)} = \begin{cases} m^{-1} \frac{\partial \phi(j \rightarrow h)}{\partial x_i(s \rightarrow r)} & \text{if } h = m_i - 1 \\ 0 & \text{else} \end{cases} $$

and the main sense of these derivatives is that they allow identifying situations in which the considered change of component state results in system degradation regardless of a concrete state (availability level) of the system.

Finally, IDPLDs of type III can be used to analyze consequences of the considered component state change on a given level of system availability and, therefore, we define them as follows:

$$ \frac{\partial \phi(h \rightarrow j \rightarrow h)}{\partial x_i(s \rightarrow r)} = \begin{cases} 1 & \text{if } \phi(s, x) \geq j \text{ and } \phi(r, x) < j \\ 0 & \text{else} \end{cases} $$

Clearly, the proposed derivatives analyzing system degradation can be nonzero for a coherent system if and only if $s > r$. Next, it is also clear that the similar IDPLDs can also be defined to analyze consequences of component improvement on system performance.

Use of IDPLDs analyzing results of component state change on system activity is presented in Figure 13. If we compare this figure with Figure 6, then we can see that the situation in which the $i$-th component degradation has no effect on system performance can be modeled by only one IDPLD while three different derivatives have to be used in the case of DPLDs (Figure 6).

Another advantage of IDPLDs is a possibility of using them in building a complex framework for the qualitative analysis of MSSs. This possibility was studied in the dissertation thesis in great detail. The main result of this study is introduction of new critical state vectors that focuses not only on a concrete component state (Table 3) but also on the whole component or the entire system. These critical state vectors for component degradation and their relations to individual IDPLDs are summarized in Table 6.

4.2 Integrated Direct Partial Logic Derivatives and Minimal Cut Vectors

IDPLDs can also be used to find MCVs and MPVs for a given level of system availability. IDPLDs of type III are most appropriate for this purpose since they are designed for analysis of consequences of component degradation (improvement) on a given level of system availability. This part of the dissertation thesis is based on our results published in papers [12, 13, 14].

Firstly, recall that MCVs for level $j$ of system availabili-
Table 6: Critical state vectors based on component degradation

<table>
<thead>
<tr>
<th>Critical State Vector</th>
<th>Definition</th>
<th>Identification based on IDPLDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Critical path vectors for system state (j) and for state (s) of the (i)-th system component</td>
<td>(\phi(s_i, x) = j) and (\phi((s - 1)_i, x) &lt; j)</td>
<td>(x_1 \leftrightarrow s) (\frac{\partial \phi(f \rightarrow)}{\partial x_i(s \rightarrow s - 1)} \leftrightarrow 1)</td>
</tr>
<tr>
<td>Critical state vectors for falling into system state (j) and for state (s) of the (i)-th system component</td>
<td>(\phi(s_i, x) &gt; j) and (\phi((s - 1)_i, x) = j)</td>
<td>(x_1 \leftrightarrow s) (\frac{\partial \phi(s \rightarrow)}{\partial x_i(s \rightarrow s - 1)} \leftrightarrow 1)</td>
</tr>
<tr>
<td>Critical path vectors for level (j) of system availability and for state (s) of the (i)-th system component</td>
<td>(\phi(s_i, x) \geq j) and (\phi((s - 1)_i, x) &lt; j)</td>
<td>(x_1 \leftrightarrow s) (\frac{\partial \phi(s \rightarrow h_{s_j})}{\partial x_i(s \rightarrow s - 1)} \leftrightarrow 1)</td>
</tr>
<tr>
<td>Critical path vectors for state (s) of the (i)-th system component</td>
<td>(\phi(s_i, x) &gt; \phi((s - 1)_i, x))</td>
<td>(x_1 \leftrightarrow s) (\frac{\partial \phi(s \rightarrow)}{\partial x_i(s \rightarrow s - 1)} \leftrightarrow 1)</td>
</tr>
<tr>
<td>Critical state vectors for degradation of system state (j) and for component (i)</td>
<td>(\phi(s_i, x) = j) and (\phi((s - 1)_i, x) &lt; j) for some (s \in {1, 2, \ldots, m_i - 1})</td>
<td>(\sum_{s=1}^{m_i-1} \frac{\partial \phi(s \rightarrow)}{\partial x_i(s \rightarrow s - 1)} \leftrightarrow 1)</td>
</tr>
<tr>
<td>Critical state vectors for falling into system state (j) and for component (i)</td>
<td>(\phi(s_i, x) = j) and (\phi((s + 1)_i, x) &gt; j) for some (s \in {0, 1, \ldots, m_i - 2})</td>
<td>(\sum_{s=1}^{m_i-1} \frac{\partial \phi(s \rightarrow)}{\partial x_i(s \rightarrow s - 1)} \leftrightarrow 1)</td>
</tr>
<tr>
<td>Critical state vectors for level (j) of system availability and for component (i)</td>
<td>(\phi((m_i - 1)_i, x) \geq j) and (\phi(0_i, x) &lt; j)</td>
<td>(\frac{\partial \phi(s_j \rightarrow h_{s_j})}{\partial x_i(m_i - 1 \rightarrow 0)} \leftrightarrow 1)</td>
</tr>
</tbody>
</table>

The critical state vectors are defined by the function \(\phi(x)\) and the partial derivatives of \(\phi\) with respect to different variables. They are useful for analyzing the system's behavior under various conditions. The table presents different scenarios for identifying critical states based on the system's structure function and the values of the variables.
and \( \partial e_x(s_1 \rightarrow s_1 + 1) \). In this case, the \( \land \)-conjunction identifies situations in which component \( i \) is in state \( s_1 \) or \( s_2 \) and its improvement causes that the system achieves at least state \( j \). Therefore, if we compute \( \land \)-conjunction of expanded integrated logic derivates \( \partial e_x(h_{c,j} \rightarrow h_{c,j})/\partial x(s \rightarrow s + 1) \) through all possible values of \( s \) for the considered component, i.e., through \( s = 0, 1, \ldots, m_i - 2 \), then we can detect all situations in which a minor improvement of the considered component results in the required change of system availability level. It follows that the computation of \( \land \)-conjunction of expressions \( \bigwedge_{x=0}^{m_i-2} \partial e_x(h_{c,j} \rightarrow h_{c,j})/\partial x(s \rightarrow s + 1) \) through all system components, i.e., through \( i = 1, 2, \ldots, n \), allows us to find state vectors at which improvement of any non-perfectly working component causes that the system reaches at least availability level \( j \), which agrees with the meaning of MCVs. This method of finding all MCVs for level \( j \) of system availability can be formalized in the form of the next algorithm:

1. Repeat the next two steps for all system components:
   1.1. Compute expanded IDPLDs 
      \( \partial e_x(h_{c,j} \rightarrow h_{c,j})/\partial x(s \rightarrow s + 1) \) for 
      \( s = 0, 1, \ldots, m_i - 2 \).
   1.2. Based on the rules in Table 7, 
      calculate \( \land \)-conjunction 
      \( \bigwedge_{x=0}^{m_i-2} \partial e_x(h_{c,j} \rightarrow h_{c,j})/\partial x(s \rightarrow s + 1) \) 
      of expanded IDPLDs computed in the previous step.
2. Using the rules defined in Table 7, 
   calculate \( \land \)-conjunction of \( \land \)-conjunctions computed in step 1, 
   and identify state vectors for which it has value 1. 
   These state vectors correspond to MCVs for 
   level \( j \) of system availability.

Clearly, if we repeat this algorithm for all possible levels of system availability, i.e., for \( j = 1, 2, \ldots, m - 1 \), then we find all MCVs of the considered system.

As in the case of BSSs, we analyzed time complexity of this algorithm based on two aspects – with respect to the number of MCVs existing in the system and with regard to the number of system components. (We have not compared this algorithm with existing ones because, according to our knowledge, the existing algorithms [23, 24, 26] are suitable for network systems only and, therefore, they cannot be applied to MSSs whose structure does not agree with a network.) The algorithm was implemented in C++ programing language and the experiments were done on a computer with CPU Intel Core i5 2.5 GHz, 4 GB RAM and Windows 7 64 bit OS. Results of the experiments are presented in Figures 14 – 16 (for simplicity, we considered only homogeneous systems). These graphs show that the algorithm speed does not depend on the number of MCVs that exist in the system but depends on the number of system components. This is more obvious in Figure 16, which shows the dependency between the number of system components and computation time of the algorithm. Furthermore, this figure also shows that the algorithm depends exponentially on the number of components states. These results are quite logical since the size of the structure function of a MSS, which is used in the computation of expanded IDPLDs and their \( \land \)-conjunction, grows exponentially with the number of system components and the number of components states.
The results of the experiments imply that the proposed algorithm is more appropriate for systems with a complicated structure, i.e., for systems containing a huge amount of MCVs, but it is not very suitable for systems consisting of lots of components or whose components have many states. Therefore, further research will be needed to propose some modifications of the algorithm that will speed it up.

Please note that the similar algorithm can also be used to find all system MPVs. However, expanded IDPLDs analyzing consequences of component degradation, i.e., \( \partial \phi(h_{x \rightarrow h})/\partial x_i (s \rightarrow s + 1) \), should be used instead of \( \partial \phi(h_{x \rightarrow h})/\partial x_i (s \rightarrow s - 1) \).

The issue of MCVs (MPVs) computation based on logic derivatives was studied in the dissertation thesis in great detail. The results of this research are presented in Table 8, which shows that there exist also other relations that can be used to find MCVs (MPVs). However, these relations are primarily suitable for symbolic manipulation with IDPLDs and, therefore, we have not considered their computer implementation. This table also introduced a concept of MCVs (MPVs) for level \( y \) of system availability and for state \( s \) of component \( i \). It is clear that these MCVs (MPVs) represent a subset of MCVs (MPVs) for a given level of system availability in which the \( i \)-th system component is in state \( s \). The notable fact is that, according to Table 8, these MCVs (MPVs) can be identified from IDPLDs directly without need to compute the expanded IDPLDs and their \( \land \)-conjunction. This will allow us to compute the FVI based on MCVs (MPVs) from IDPLDs directly without a priori knowledge of MCVs (MPVs).

At the end of this part, please note that the previous paragraphs indicate that we showed in the dissertation thesis that logical differential calculus can be applied to a MSS of any type (not only to the special type of homogeneous systems). This result allowed us to propose a complex framework for the qualitative analysis of MSSs that combine the concept of component criticality with the concept of MCVs (MPVs). Based on this, we can state that logical differential calculus can be seen as a tool that allows us to perform the complex qualitative analysis of MSSs (Figure 17).

4.3 Integrated Direct Partial Logical Derivatives and Importance Analysis

Based on the relation between IDPLDs and critical state vectors, a complex framework for importance analysis of MSSs was developed in the dissertation thesis. This framework allows identifying components or their states that have the most influence on the entire system or on a specified system state (availability level). For this purpose, definitions of the SI, BI, CI and FVI based on IDPLDs were proposed in the thesis. We also showed that our definitions of the SI and BI can be viewed as a generalization of those considered in works \([1, 6, 21, 30]\). As an example of our approach, let us consider Table 9, which contains all SI measures analyzing impact of a minor degradation of a system component proposed in the thesis. The first part of this table contains SI measures that identify coincidence between degradation of a given component state and a given system state (availability level). In the second part, the SI for finding component state whose degradation results in system degradation most frequently is presented. The third part of Table 9 includes SI measures that analyze dependency between component degradation and degradation of a given system state (availability level). These SI measures can be used to find components that are most important for the considered system state (availability level) from topological point of view. Finally, the SI analyzing the total topological importance of a given component on the system is presented in the last row of the table. Furthermore, relationships between individual types of the SI measures were considered in the thesis. It was shown that all SI measures from the second, third, and fourth part of the table can be computed directly from the SI analyzing the coincidence between degradation of a given component state and a given system state (availability level). IDPLDs were also used to propose the similar definitions for computation of the BI, CI, and FVI measures.

One of the principal contributions of this part is developing new formulae for calculation of the FVI that allow calculating this IM directly from IDPLDs without a priori knowledge of the MCVs. These formulae were derived in the similar way as in the case of the FVI for BSSs and, therefore, the FVI of a given component state for a given level of system unavailability can be computed in the following way:

\[
FV_{i,s} = \Pr \left\{ \text{MNF} \left( \frac{\partial \phi(h_{x \rightarrow h})}{\partial x_i (s \rightarrow s - 1)} \right) \leftrightarrow 1 \right\} \frac{p_{i,s-1}}{U(p)}
\]

where MNF(\(\cdot\)) is the minimal negative function for a function that can be interpreted as a function with Boolean-valued output (note that DPLDs and IDPLDs have only two possible values – 0 and 1, therefore, they can be viewed as functions with Boolean valued output), \(p_{i,s-1}\) denotes the probability that component \( i \) is in state \( s - 1 \), i.e. a minor degradation of state \( s \) of component \( i \) has occurred, and \( U(p) \) is system unavailability computed with respect to system state \( j \).

At the end of this section, please note that the proposed FVI estimates the probability that a degradation of a given state of a given component contributes to system unavailability, which is in agreement with the original FVI proposed for BSSs. We mention this fact because there also exist other definitions of the FVI for MSS, e.g., \([15, 18, 33]\), but they are more similar to the CI than to the FVI. Therefore, one of the main contributions of our work is the extension of the original FVI from BSSs on MSSs of any type.

5. Conclusion

In the dissertation thesis, the problem of developing new algorithms for reliability analysis of BSSs and MSSs represented by the structure function was considered. New algorithms for identification of MCVs and MPVs for systems of any type were developed. The algorithms are based on the relation between MCVs (MPVs) and logical differential calculus that was identified in this work. New algorithms for importance analysis of non-homogeneous MSSs that are based on finding critical state vectors using logical differential calculus were proposed also in this work. These results were achieved by successful solving the next problems:

- detailed analysis of existing methods and concepts of reliability analysis:
Table 8: Minimal cut/path vectors for system availability level and their relation to (expanded) IDPLDs of type III

<table>
<thead>
<tr>
<th>Concept</th>
<th>Definition</th>
<th>Identification based on IDPLDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimal cut vectors for level ( j ) of system availability</td>
<td>( \phi(x) &lt; j ) and ( \phi(y) \geq j ) for any ( y &gt; x )</td>
<td>( \bigwedge_{i=1}^{n} { x_i \leftrightarrow m_i-1 } \vee \left{ \bigwedge_{i=1}^{n} { x_i \leftrightarrow s } \phi \left( \frac{h_{x_i} - h_{y_i}}{x_i - s} \right) \right} \bigwedge_{i=1}^{n} { x_i \geq s } )</td>
</tr>
<tr>
<td>Minimal cut vectors for level ( j ) of system availability and for state ( s ) of component ( i )</td>
<td>( \phi(s_i, x) &lt; j ) and ( \phi(y) \geq j ) for any ( y &gt; (s_i, x) )</td>
<td>( \max_{x \in M(\phi(x))} \left{ \left[ { x_i \leftrightarrow s } \phi \left( \frac{h_{x_i} - h_{y_i}}{x_i - s} \right) \right] \right} )</td>
</tr>
<tr>
<td>Minimal path vectors for level ( j ) of system availability</td>
<td>( \phi(x) \geq j ) and ( \phi(y) &lt; j ) for any ( y &lt; (s_i, x) )</td>
<td>( \bigwedge_{i=1}^{n} { x_i \leftrightarrow 0 } \vee \left{ \bigwedge_{i=1}^{n} { x_i \leftrightarrow s } \phi \left( \frac{h_{x_i} - h_{y_i}}{x_i - s} \right) \right} \bigwedge_{i=1}^{n} { x_i &gt; 0 } )</td>
</tr>
<tr>
<td>Minimal path vectors for level ( j ) of system availability and for state ( s ) of component ( i )</td>
<td>( \phi(s_i, x) \geq j ) and ( \phi(y) &lt; j ) for any ( y &lt; (s_i, x) )</td>
<td>( \min_{x \in M(\phi(x))} \left{ \left[ { x_i \leftrightarrow s } \phi \left( \frac{h_{x_i} - h_{y_i}}{x_i - s} \right) \right] \right} )</td>
</tr>
</tbody>
</table>

*note: \( M(\phi(x)) = \{0, 1, ..., m_1 \} \times \{0, 1, ..., m_2 \} \times ... \times \{0, 1, ..., m_n \} \)

Figure 17: Logical differential calculus as a unifying tool in the qualitative analysis of MSSs
Table 9: Structural importance measures based on component degradation

<table>
<thead>
<tr>
<th>Structural Importance</th>
<th>Definition</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>The SI of given component state and for given system state</td>
<td>$SI_{i,s}^{j} = TD \left( \frac{\partial \phi(j \wedge j)}{\partial x_{i}(s \rightarrow s - 1)} \right)$</td>
<td>A relative number of situations in which state $s$ of component $i$ is critical for degradation of state $j$ of the system.</td>
</tr>
<tr>
<td>The SI of given component state and for given level of system availability</td>
<td>$SI_{i,s}^{j} = TD \left( \frac{\partial \phi(j \wedge h_{c,j})}{\partial x_{i}(s \rightarrow s - 1)} \right)$</td>
<td>A relative number of situations in which state $s$ of component $i$ is critical for degrading level $j$ of system availability.</td>
</tr>
<tr>
<td>The SI of given component state</td>
<td>$SI_{i,s} = TD \left( \frac{\partial \phi(j \wedge s)}{\partial x_{i}(s \rightarrow s - 1)} \right)$</td>
<td>A relative number of situations in which state $s$ of component $i$ is critical for system degradation.</td>
</tr>
<tr>
<td>The SI of given component state and for given system state</td>
<td>$SI_{i,s}^{j} = TD \left( \frac{\sum_{s=1}^{m_{i}-1} \frac{\partial \phi(j \wedge s)}{\partial x_{i}(s \rightarrow s - 1)}}{m_{i} - 1} \right)$</td>
<td>A relative number of situations in which component $i$ is critical for degradation of state $j$ of the system.</td>
</tr>
<tr>
<td>The SI of given component for given system availability level</td>
<td>$SI_{i}^{j} = TD \left( \frac{\partial \phi(h_{c,j} \wedge h_{c,j})}{\partial x_{i}(m_{i} - 1 \rightarrow 0)} \right)$</td>
<td>A relative number of situations in which component $i$ is critical for degradation of level $j$ of system availability.</td>
</tr>
<tr>
<td>The total SI of given component</td>
<td>$SI_{i}^{\gamma} = \frac{\sum_{s=1}^{m_{i}-1} SI_{i,s}^{j}}{m_{i} - 1}$</td>
<td>A relative number of situations in which component $i$ is critical for system degradation.</td>
</tr>
</tbody>
</table>

— it was presented that a lot of methods and approaches exist in reliability analysis, especially in the analysis of MSSs;
— it was showed that one IM has several meanings;
• development of universal mathematical background for calculation of MCVs (MCVs) and MPVs (MPVs) for BSSs and MSSs:
  - new types of logic derivatives were proposed:
    * concept of expanded DPBDs and DPLDs for reliability analysis of BSSs and MSSs respectively was introduced;
    * the new type of DPLDs that were named IDPLDs for analysis of non-homogeneous MSSs was defined;
  - the correlation between IDPLDs, critical state vectors, and MCVs (MPVs) of MSSs was presented. This correlation indicates that logical differential calculus can be viewed as a tool unifying several methods used in reliability analysis of MSSs (Figure 17);
  - new algorithms for identification of MCVs and MPVs of BSSs and MSSs that are based on the concepts of expanded derivatives and IDPLDs were developed;
• creating new algorithms for calculation of measures for system reliability/availability estimation:
  - new methods for computation of the SI, BI, CI, and FVI measures were developed based on IDPLDs for MSSs. These methods allow identifying:
    * components with the most influence on system degradation/ improvement;
    * components with the most influence on degradation/ improvement of a concrete system state (availability level);
    * component states with the most impact on system degradation/ improvement;
    * component states with the most impact on degradation/ improvement of a specified system state (availability level);
  - it was showed that the proposed definitions of IMs combined several approaches used in importance analysis of MSSs;

The principal contribution of this work is development of new mathematical method for importance analysis that can be used for BSSs and MSSs without any significant modifications. It is important that the new technique for calculation of the FVI for MSSs was proposed based on this method. According to our knowledge, it is the
first technique that calculates the FVI whose meaning corresponds to the original FVI that has been proposed for BSSs.

The future work should focus on finding some ways of how to make the algorithm for identification of MCVs (MPVs) more efficient for MSSs with a lot of components or lot of components states since its analysis showed that it is not very suitable for such systems. Next, the idea of expanded DPBDs, DPLDs, and $\wedge$-conjunction was introduced in this work. This idea was primarily aimed to propose a simple implementation of the algorithm for finding all system MCVs (MPVs). However, our current research indicates that it might be useful in identification of prime implicants or prime implicants of noncoherent systems. Some ideas of how to use logical differential calculus in reliability analysis of such systems have been presented in papers [9, 29]. These works have shown that logical differential calculus is very intuitive for noncoherent systems and, therefore, it could be used as a tool that extends some techniques from the analysis of coherent systems on noncoherent ones.

References


Selected Papers by the Author


