

Elena ZAITSEVA
Vitaly LEVASHENKO
Jozef KOSTOLNY

APPLICATION OF LOGICAL DIFFERENTIAL CALCULUS AND BINARY DECISION DIAGRAM IN IMPORTANCE ANALYSIS

ZASTOSOWANIE LOGICZNEGO RACHUNKU RÓŻNICZKOWEGO ORAZ BINARNEGO DIAGRAMU DECYZYJNEGO W ANALIZIE WAŻNOŚCI

System availability evaluation includes different aspects of system behaviour and one of them is the importance analysis. This analysis supposes the estimation of system component influence to system availability. There are different mathematical approaches to the development of this analysis. The structure function based approach is one of them. In this case system is presented in form of structure function that is defined the correlation of system availability and its components states. Structure function enables one to represent mathematically a system of any complexity. But computational complexity of structure function based methods is time consuming for large-scale system. Decision of this problem for the calculation of importance measures can be realized based on application of two mathematical approaches. One of them is Direct Partial Boolean Derivative. New equations for calculating the importance measures are obtained in terms of these derivatives. Other approach is Binary Decision Diagram (BDD), which supports efficient manipulation of Boolean algebra. In this paper new algorithms for calculating of importance measures by Direct Partial Boolean Derivative based on BDD are proposed. The experimental results of comparison these algorithms with other show the efficiency of new algorithms for calculating Direct Partial Boolean Derivative and importance measures.

Keywords: decision diagram, importance analysis, logical differential calculus, reliability engineering.

Ocena gotowości systemu, analiza czułości, miary ważności oraz optymalna konstrukcja to istotne zagadnienia, które stały się obiektem badań z zakresu inżynierii niezawodności. Istnieją różne podejścia matematyczne do owych problemów. Jednym z nich jest podejście oparte na funkcji struktury. Funkcja struktury umożliwia analizę systemów o wszelkim stopniu złożoności. Jednakże, w przypadku sieci o dużej skali, złożoność obliczeniowa metod opartych na funkcji struktury sprawia, że metody te są czasochłonne. W przedstawionej pracy proponujemy wykorzystanie dwóch metod matematycznych analizy ważności. Pierwszą z nich jest bezpośrednia cząstkowa pochodna boole'owska, w kategoriach której opracowano nowe równania do obliczania miar ważności. Drugą jest binarny diagram decyzyjny, który wspiera efektywną manipulację na wyrażeniach algebry Boole'a. W artykule zaproponowano dwa algorytmy służące do obliczania bezpośredniej cząstkowej pochodnej boole'owskiej w oparciu o binarny diagram decyzyjny funkcji struktury. Wyniki eksperymentów wykazują skuteczność nowo opracowanych algorytmów w obliczaniu bezpośredniej cząstkowej pochodnej boole'owskiej oraz miar ważności.

Słowa kluczowe: diagram decyzyjny, analiza ważności, logiczny rachunek różniczkowy, inżynieria niezawodności.

1. Introduction

A mathematical representation and description or creation of mathematical model of the initial object is an important step in reliability/availability analysis. There are some types of such mathematical models in reliability engineering. As a rule the mathematical model type is conditioned by mathematical method that is used for system availability or reliability estimates. One of mathematical model is the structure function of a system that is defined by one-to-one mapping of system state and states of the system components. The structure function can be interpreted as Boolean function if the initial system has two possible states as functioning and failure only and the system analysed in the stationary state [4, 32]. Therefore mathematical methods of Boolean algebra can be used for the structure function analysis and as result for estimation of system availability. There are a lot of algorithms for the calculation of special indices and measures in reliability engineering based on Boolean algebra methods. Some of them are *Importance Measures* (IMs).

IMs were introduced by Birnbaum in [6]. These measures express the contribution of a component state to system reliability/availability. At the present time there are different types of IMS that allows investigating different aspects of the influence of appointed component state change to behaviour of system reliability or availability. Every system component has a measure of its importance to the system functioning and failure according to values of IMs, that are calculated for every component. IMs have been widely used for identifying a system's weakness and supporting system improvement activities from design perspective. With the importance values of all components, proper actions can be taken on the weakest component to improve system availability at minimal cost or effort.

Detailed analysis and comparison of IMs are presented, for example, in papers [8, 13, 19, 25]. Different mathematical approaches are used and developed for calculating IMs, some of which are Boolean methods [3, 8, 5], Markov model [12], minimal cuts/paths set methods [11, 19], and Logical Differential Calculus [35]. In this paper, we develop a method for computing IMs based on Logical Differential Calculus with the application of *Binary Decision Diagram* (BDD).

Logical Differential Calculus is a mathematical approach that permits to investigate changes in Boolean function depending on changes of its variables values [1, 33]. Therefore, this tool can be used to evaluate the influence of every system component state change on the system performance level [24, 32, 37]. In this paper the Direct Partial Boolean Derivatives are used for the development of algorithms for the estimation of system availability. However, the Logical Differential Calculus methods and Direct Partial Boolean Derivatives in particular are characterized by high computational complexity that increases depending on the number of system components. In Boolean algebra there are some tools to decide this problem and one of them is BDD. A BDD is one of the more effective methods for the representation of a Boolean function of large dimensions[1].

A BDD is used not only in Boolean algebra. A BDD is widely and efficiently used in reliability analysis[14, 15, 16, 28]. Methods for transformation of a Fault Tree to the ordered BDD (OBDD) are proposed in papers [17, 28]. These methods permit to substantiate the correct application of OBDD in reliability analysis. The development of these methods and OBDD use for other problems in reliability analysis has been proposed in a lot of papers. For example, new algorithm for the manipulation of series and parallel systems based on BDD are investigated in [31]. The paper authors [18, 21, 22, 23] consider the OBDD application for reliability analysis of dynamic, phased-mission and networking systems.

The computation of IMs by BDD is considered in papers [8] and[14]. Authors of paper [14] investigated IMs definitions in terms of OBDD. In papers [14] and [22] similar algorithms for the calculation of IM as Birnbaum Importance are proposed based on equal interpretation of this measure definition/equation. The Birnbaum Importance is calculated directly using the OBDD. But these algorithms can be used for the OBDD only.

The combination of two mathematical tools for IMs computation. In this paper we develop the results that have been considered in [31, 37] for BDD. The strong evidence of Logical Differential Calculus application in importance analysis is considered in [37] and new equations for computation of the IMs as Structural Importance, Birnbaum Importance and Criticality Importance in the terms of Logical Differential Calculus have been proposed in the paper [37]. In this paper we use these equations for the development a new BDD-based approach. We compare of IMs values calculated by the new algorithms and classical definition of IMs (these definition are in [3] for Structural Importance and in [6, 12] for Birnbaum Importance). New algorithms allow obtaining the IMs values that are equal with the IMs values computed by the definitions in [3, 6, 12]. Therefore the principal step of the new approach for the calculation of IM presented in this paper is the application of Logical Differential Calculus, in particular Direct Partial Boolean Derivatives that have been considered in [1, 33, 37]. Two algorithms for the calculation of Direct Partial Boolean Derivatives are proposed in this paper and compared with alternative algorithms.

The remainder of the work is organized as follows. Section 2 introduces concepts of the system structure function, Logical Differential Calculus and BDD. In Section 3 the calculation of IMs by Logical Differential Calculus is described. In this section the definitions of IMs in terms of Logical Differential Calculus are considered. Two algorithms for the IMs quantification based on BDD are considered in this section too. These algorithms are founded on two mathematical tools: Logical Differential Calculus and BDD. A benchmark study is presented to illustrate the proposed algorithms.

2. Mathematical background

2.1. The system structure function

Consider the mathematical model of the system of n components. The component state is denoted as x_i ($i = 1, \dots, n$). The value $x_i = 1$ corresponds to the operable state of the i -th component, and $x_i = 0$, to its failure. Every system component is characterized by probabilities of its states. The probability of i -th component failure is:

$$q_i = \Pr\{x_i = 0\}. \quad (1)$$

Then, the probability of operability of the i -th component is defined as:

$$p_i = \Pr\{x_i = 1\}. \quad (2)$$

Consider the system in stationary state. The system availability in stationary state is represented mathematically by the structure function[4, 27]:

$$\phi(x) = \phi(x_1, \dots, x_n): \{0, 1\}^n \rightarrow \{0, 1\}, \quad (3)$$

where the vector $x = (x_1, \dots, x_n)$ is the vector of the system components states.

Note, the structure function (3) can be interpreted as Boolean function. As a rule, a coherent system is investigated in reliability engineering. Such system will be considered in this paper. The structure function of a coherent system has the following assumptions [6]:

- The system and its components have two states: up (working) and down (failed);
- All system components are relevant to the system;
- The system structure function is monotone non-decreasing: $\phi(x_1, \dots, 1, \dots, x_n) \neq \phi(x_1, \dots, 0, \dots, x_n)$;
- The failure and repair rate of the components are constant;
- Repaired components are as good as new.

Principal indices and measures of the system can be calculated based on the structure function. The one of them is the system unavailability that is the probability of the system failure[27]:

$$U = \Pr\{\phi(x) = 0\} \quad (4)$$

and the system availability that is probability of the system functioning state:

$$A = \Pr\{\phi(x) = 1\}. \quad (5)$$

The system unavailability and availability correlate by the equation:

$$U + A = 1$$

The system unavailability (4) and availability (5) are calculated based on every system component state x_i probabilities (1) and (2). For example, the structure function of the system in Fig.1 of 4 components is defined as:

$$\phi(x) = \text{AND}(x_1, \text{OR}(x_2, \text{AND}(x_3, x_4))) = x_1 \wedge (x_2 \vee (x_3 \wedge x_4)). \quad (6)$$

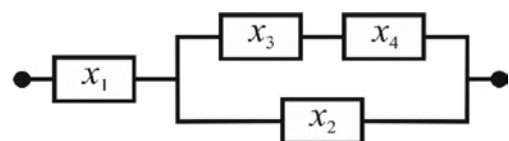


Fig. 1. The series-parallel system example

The system availability is calculated according to (5) taking into account the truth table of the structure function or its canonical DNF:

$$\phi(x) = x_1\bar{x}_2x_3x_4 \vee x_1x_2\bar{x}_3\bar{x}_4 \vee x_1x_2\bar{x}_3x_4 \vee x_1x_2x_3\bar{x}_4 \vee x_1x_2x_3x_4$$

and the system availability is calculated according to (5):

$$A = p_1q_2p_3p_4 + p_1p_2q_3q_4 + p_1p_2q_3p_4 + p_1p_2p_3q_4 + p_1p_2p_3p_4 = p_1p_3p_4 + p_1p_2 - p_1p_2p_3p_4 \quad (7)$$

The system unavailability (4) and availability (5) are important measures, but these measures don't allow to investigate all aspect of the system availability. Therefore other measures must be defined that permit to estimate of the system behavior in point of view of reliability analysis. It can be, for example, IMs or other indices. The use of these measures and indices are caused the development of special methods and algorithms for these estimations. There are different mathematical approaches that are used for development and calculation of reliability indices and measures. And one of them are based on Boolean algebra, because the structure function (3) can be interpreted as a Boolean function. Two mathematical tools will be used in this paper: BDD and Logical Differential Calculus.

A BDD is a graphical form of a Boolean function [1] that is efficiency for the representation and analysis of a Boolean function of large dimension. The second tool is the Logical Differential Calculus [2, 10, 33]. Mathematical methods of Logical Differential Calculus permit to investigate the influence of a variable value change to the Boolean function value. In papers[24, 32, 37] some methods and algorithms of applying Logical Differential Calculus in reliability engineering are considered. In this case the system availability agrees with a Boolean function value and the *i*-th component state is interpreted as the *i*-th variable value. Consider these tools below.

2.2. Logical Differential Calculus in Reliability Analysis

The application of Logical Differential Calculus in reliability analysis has been considered in [24, 32, 37]. Authors of these papers use Boolean Derivative (Boolean Difference). In this paper we propose to use other derivative that allows analysis of the influence of variable value change to a function value in more detail. It is Direct Partial Boolean Derivative.

Definition 1. The Direct Partial Boolean Derivative of the function $\phi(x)$ with respect to variable x_i reflects the fact of changing the function from j to \bar{j} when the value of variable x_i changes from s to \bar{s} [34]:

$$\partial\phi(j \rightarrow \bar{j})/\partial x_i(s \rightarrow \bar{s}) = \begin{cases} 1, & \text{if } \phi(s_i, x) = j \text{ and } \phi(\bar{s}_i, x) = \bar{j} \\ 0, & \text{other} \end{cases} \quad (8)$$

where $\phi(s_i, x) = \phi(x_1, \dots, x_{i-1}, s, x_{i+1}, \dots, x_n), s, j \in \{0, 1\}; =1-j, =1-s;$ and \equiv is the symbol of equivalence operation.

Direct Partial Boolean Derivatives allows mathematical representation of a system fault that is caused by the *i*-th system component failure by derivatives $\partial\phi(1 \rightarrow 0)/\partial x_i(1 \rightarrow 0)$ and $\partial\phi(1 \rightarrow 0)/\partial x_i(0 \rightarrow 1)$.

Derivatives $\partial\phi(0 \rightarrow 1)/\partial x_i(0 \rightarrow 1)$ and $\partial\phi(0 \rightarrow 1)/\partial x_i(1 \rightarrow 0)$ describe the system renewal if the *i*-th component repaired

Direct Partial Boolean Derivative of a coherent system has the following properties for monotone function [29, 37]:

$$\partial\phi(1 \rightarrow 0)/\partial x_i(0 \rightarrow 1) = \partial\phi(0 \rightarrow 1)/\partial x_i(1 \rightarrow 0) = 0, \quad (9)$$

$$\partial\phi(0 \rightarrow 1)/\partial x_i(0 \rightarrow 1) = \overline{\partial\phi(1 \rightarrow 0)/\partial x_i(1 \rightarrow 0)}. \quad (10)$$

Because the structure function of a coherent system $\phi(x)$ is monotone (assumption c) we will investigate the derivatives of a system fault $\partial\phi(1 \rightarrow 0)/\partial x_i(1 \rightarrow 0)$ and renewal $\partial\phi(0 \rightarrow 1)/\partial x_i(0 \rightarrow 1)$.

Therefore the denotation will be used for these derivatives below:

$$\partial\phi(x)/\partial x_i = \partial\phi(0 \rightarrow 1)/\partial x_i(0 \rightarrow 1) = \overline{\partial\phi(1 \rightarrow 0)/\partial x_i(1 \rightarrow 0)}$$

and the Direct Partial Boolean Derivative $\partial\phi(x)/\partial x_i$ is calculated as[34, 37]:

$$\partial\phi(x)/\partial x_i = \phi(1_i, x) \wedge \overline{\phi(0_i, x)}. \quad (11)$$

Direct Partial Boolean Derivative (11) permits us to determine the boundary states of a system for which the change of the *i*-th component state cause the change of the system availability. Nonzero values of the derivatives indicate the system states (the vector of system components states) for which the *i*-th component fault causes the system failure (Fig. 2).

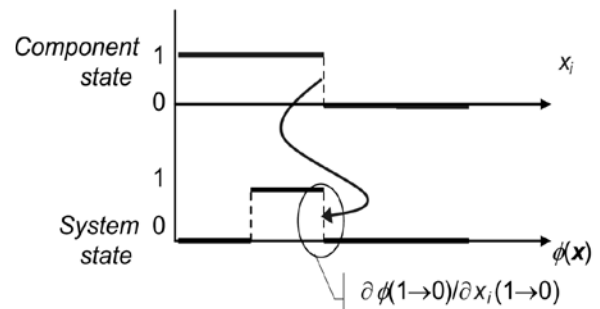


Fig. 2. The interpretation of Direct Partial Boolean Derivative

For example, consider the system in Fig.1 with the structure function (6). The structure function truth table of this system and Direct Partial Boolean Derivatives with respect to variable x_i ($i = 1, 2, 3, 4$) are shown in Table 1. Note, the symbol “*” in this table marks the state vectors for which derivatives don't exist. The nonzero values of derivatives indicate the boundary states of the system. These states are defined by the vector states:

- for the first component $x = (\underline{1 \leftrightarrow 0}, 0, 1, 1), x = (\underline{1 \leftrightarrow 0}, 1, 0, 0), x = (\underline{1 \leftrightarrow 0}, 1, 0, 1), x = (\underline{1 \leftrightarrow 0}, 1, 1, 0), x = (\underline{1 \leftrightarrow 0}, 1, 1, 1);$
- for the second component $x = (1, \underline{1 \leftrightarrow 0}, 0, 0), x = (1, \underline{1 \leftrightarrow 0}, 0, 1), x = (1, \underline{1 \leftrightarrow 0}, 1, 0);$
- for the third component $x = (1, 0, \underline{1 \leftrightarrow 0}, 1);$
- for the fourth component $x = (1, 0, 1, \underline{1 \leftrightarrow 0}).$

Direct Partial Boolean Derivative with respect to the first variable x_1 has 5 nonzero values that indicate boundary states. In this case, as state change of the first component influences the system availability. Vector state $x = (\underline{1 \leftrightarrow 0}, 0, 1, 1)$ indicates that the first component failure causes the system failure if the second component isn't functioning and the third and fourth components are working. Vector states $x = (\underline{1 \leftrightarrow 0}, 1, 0, 0), x = (\underline{1 \leftrightarrow 0}, 1, 0, 1), x = (\underline{1 \leftrightarrow 0}, 1, 1, 0), x = (\underline{1 \leftrightarrow 0}, 1, 1, 1)$ define the system failure for any states of the third and the fourth components if the second component is functioning. Other derivatives in Table 1 have similar meanings.

The algorithms for calculating Direct Partial Boolean Derivatives have been considered in numerous papers. Trivial algorithm has been considered by Akers in [2]. This result has been developed by Bochmann, Posthoff and Steinbach [7, 26]. In paper [30] a parallel version of these algorithms has been proposed. The calculation of Boolean Derivative (Boolean Difference) based on the BDD has been investigated in [24]. However, this algorithm cannot be used for Direct

Table 1. The structure function and Direct Boolean Derivatives of the series-parallel system (Fig. 1)

$x_1 x_2 x_3 x_4$	$\phi(\mathbf{x})$	$\partial\phi(\mathbf{x})/\partial x_1$	$\partial\phi(\mathbf{x})/\partial x_2$	$\partial\phi(\mathbf{x})/\partial x_3$	$\partial\phi(\mathbf{x})/\partial x_4$
0 0 0 0	0	*	*	*	*
0 0 0 1	0	*	*	*	0
0 0 1 0	0	*	*	0	*
0 0 1 1	0	*	*	0	0
0 1 0 0	0	*	0	*	*
0 1 0 1	0	*	0	*	0
0 1 1 0	0	*	0	0	*
0 1 1 1	0	*	0	0	0
1 0 0 0	0	0	*	*	*
1 0 0 1	0	0	*	*	0
1 0 1 0	0	0	*	0	*
1 0 1 1	1	1	*	1	1
1 1 0 0	1	1	1	*	*
1 1 0 1	1	1	1	*	0
1 1 1 0	1	1	1	0	*
1 1 1 1	1	1	0	0	0

Partial Boolean Derivatives calculation. Therefore the algorithm for the calculation of Direct Partial Boolean Derivatives based on BDD isn't developed.

2.3. Binary Decision Diagram

Consider some background of BDD before the development of algorithm for the calculation of Direct Partial Boolean Derivatives based on BDD.

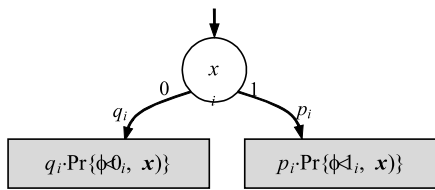
BDD is a widely used tool for reliability analysis. Some methods for reliability analysis based on this tool are discussed in papers [14, 15, 16]. BDD is based on a disjoint decomposition of Boolean function called the Shannon expansion [14]. This expansion for the structure function (3) can be defined as:

$$\phi(\mathbf{x}) = x_i \wedge \phi(1_i, \mathbf{x}) \vee \bar{x}_i \wedge \phi(0_i, \mathbf{x}) \quad (12)$$

In order to express Shannon decomposition concisely, the if-then-else (ite) format[9] is defined as:

$$\phi(\mathbf{x}) = ite(x_i=0, \phi(0_i, \mathbf{x}), \phi(1_i, \mathbf{x})) \quad (13)$$

Non-terminal node with the component state probabilities



Rules for BDD manipulation

Multiplication Addition Multiplication and Addition

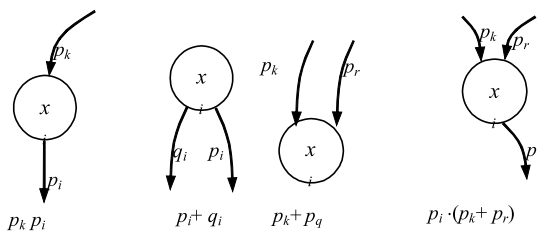
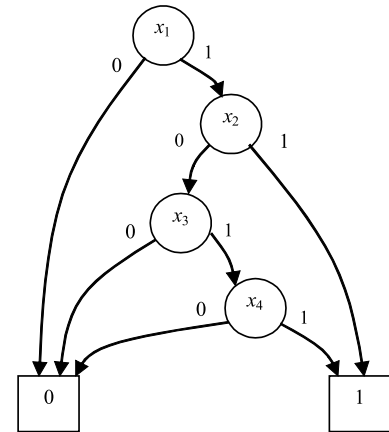


Fig. 3. Interpretation of component state and basic computational rules in BDD

BDD of system



The pseudo-code of the BDD

```

(x) = ite(x1, 0, L)
L = ite(x2, M, 1)
M = ite(x3, 0, N)
N = ite(x4, 0, 1)

or

(x) = ite(x1, 0, ite(x2, ite(x3, 0,
ite(x4, 0, 1)), 1))
    
```

Fig. 4. Graphical and software implementation of the BDD of system

A BDD is a directed acyclic graph of a Boolean function representation [9]. For structure function (3), this graph has two terminal nodes, labelled 0 and 1. Each non-terminal node is labelled with a structure function variable x_i and has two outgoing edges. The left edge is labelled "0" and represents the fail state of system component. The other outgoing edge is labelled "1" and represents the operational state of a system component.

Terminal nodes of the BDD correspond to the system state. Non-terminal node outgoing edges are interpreted as component states. The probabilistic interpretation of the system assumes that every edge from the node to which the variable x_i is assigned with the label s_i is marked by the i -th component state probability p_i or q_i (Fig. 3). The node in this diagram assigned with the i -th variable x_i , and the outgoing edges corresponding to $\phi(1_i, \mathbf{x})$ and $\phi(0_i, \mathbf{x})$ that correspond to the Shannon decomposition (12). This equation in arithmetic form is the following:

$$\phi(\mathbf{x}) = x_i \cdot \phi(1_i, \mathbf{x}) + \bar{x}_i \cdot \phi(0_i, \mathbf{x}) - (x_i \cdot \phi(1_i, \mathbf{x})) \cdot (\bar{x}_i \cdot \phi(0_i, \mathbf{x})) = x_i \cdot \phi(1_i, \mathbf{x}) + \bar{x}_i \cdot \phi(0_i, \mathbf{x})$$

Node probability is calculated using the equation [14]:

$$\Pr\{\phi(\mathbf{x})\} = \Pr\{x_i \cdot \phi(1_i, \mathbf{x})\} + \Pr\{\bar{x}_i \cdot \phi(0_i, \mathbf{x})\} = p_i \cdot \Pr\{\phi(1_i, \mathbf{x})\} + q_i \cdot \Pr\{\phi(0_i, \mathbf{x})\} \quad (14)$$

Note that the paths from the root to the terminal node in a BDD are mutually disjoint. Therefore the system availability can be calculated based on (12) and (14) for the system that is represented in form of the BDD:

$$A = \Pr\{\phi(\mathbf{x}) = 1\} = p_i \cdot \Pr\{\phi(1_i, \mathbf{x}) = 1\} + q_i \cdot \Pr\{\phi(0_i, \mathbf{x}) = 1\} \quad (15)$$

and conforms with the probability of the sum of paths from the root node to the terminal node "1". Rules for this calculation are presented

in Fig. 3. The system unavailability is calculated similarly but paths from the root node to the terminal node labelled “0” are analysed in this case.

For example, consider a system 2-out-of-3 with structure function (6). The system is defined by the BDD (Fig.4). The calculation of the system availability and unavailability is implemented based on two sub-diagrams. Every diagram is the sum of paths from the BDD top to one of the terminal nodes (Fig.5).

The sub-diagram for the system unavailability calculation **The sub-diagram for the system availability calculation**

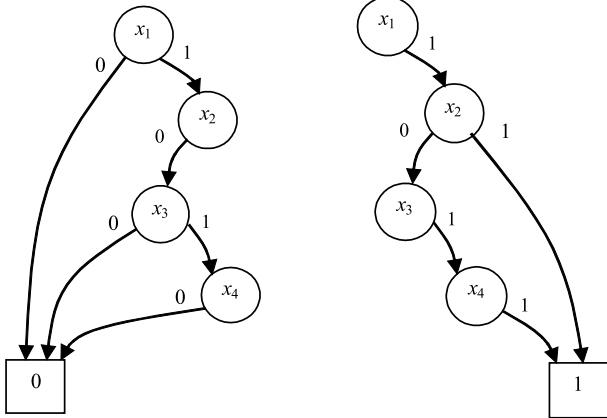


Fig. 5. Sub-diagrams for the system availability and unavailability calculation

According to the sub-diagrams in Fig.5 the system availability and unavailability are calculated as:

$$A = p_1(p_2 + q_2p_3p_4) = p_1p_2 + p_1p_3p_4 + p_1p_2p_3p_4 \quad (16)$$

$$U = q_1 + p_1q_2(q_3 + p_3q_4). \quad (17)$$

The system availability (16) is equal to the system availability (7) that was defined according to typical rules. Therefore, BDD can be used for the calculation of the system’s availability and unavailability.

3. Importance Analysis

System’s unavailability (4) and availability (5) are widely used measures in reliability engineering, but these measures do not enable the analysis of a change of the system availability depending on a change of the component states. There are measures for estimating the influence of component states’ changes on system availability called IMs. Importance analysis allows the estimation of influence of every system component breakdown to the system failure. This paper provides new algorithms for importance analysis based on Direct Partial Boolean Derivative and BDD of structure function.

3.1. Importance Measures

Consider some of IMs and calculation algorithms based on Direct Partial Boolean Derivative.

Structural Importance (SI). The SI is one of the simplest measures in importance analysis and this measure focuses on the topological aspects of a system. According to the definition in papers [3] this measure determines the proportion of working states of a system in which the working of the i -th component makes the difference between system failure and operation:

$$SI_i = \frac{\rho_i}{2^{n-1}} \quad (18)$$

where ρ_i is the number of system states when the change in component state results in system failure.

Table 2. SI and MSI indices for the system in Fig. 1

i	ρ_i	$\rho_{i=1}$	SI_i	MSI_i
1	5	5	0.625	1.000
2	3	4	0.250	0.750
3	1	2	0.125	0.500
4	1	2	0.125	0.500

where $\rho_{i=1}$ is the number of system states for which $\phi(1_i, \mathbf{x}) = 1$ (it is defined by the structure function (3)).

The number ρ_i in [3] is defined by special analysis of values of the structure function, but according to the definition 1 this number can be calculated as the number of nonzero values of the Direct Partial Boolean Derivatives (11) $\partial\phi(\mathbf{x})/\partial x_i$.

There is one more definition of SI [37]. It is **Modified Structural Importance (MSI)** that represents the influence of the i -th system component failure to system failure:

$$MSI_i = \frac{\rho_i}{\rho_{i=1}} \quad (19)$$

where $\rho_{i=1}$ is the number of system states for which $\phi(1_i, \mathbf{x}) = 1$ (it is defined by the structure function (3)).

The values of SI (18) and MSI (19) are presented in Table 2. According to this Table the first system component has maximal influence to the system availability in terms of the system’s topology. The values of the MSI indices show that the first component is dominant in terms of system topology because only this component’s fault causes the system’s failure. The breakdown of the third or the fourth component has minimal influence to the system failure according to values of SI and MSI (Table 2).

Birnbaum Importance (BI). The BI is one of basic IMs and this measure is defined as the probability that the system is sensitive to inoperative state of the i -th system component[19]:

$$BI_i = \Pr\{\varphi(0_i, \mathbf{x}) = 0\} - \Pr\{\varphi(1_i, \mathbf{x}) = 0\}.$$

In paper[37] new equation for the BI calculation has been proposed based on Direct Partial Boolean Derivatives:

$$BI_i = \Pr\{\partial\varphi(\mathbf{x})/\partial x_i = 1\}. \quad (20)$$

According to [37] this equation is proofed as:

$$\begin{aligned} I_B(x_i) &= \Pr\{(\varphi(1_i, \mathbf{x}) \wedge \overline{\varphi(0_i, \mathbf{x})}) = 1\} = \Pr\{\varphi(1_i, \mathbf{x}) = 1\} \cdot \Pr\{\overline{\varphi(0_i, \mathbf{x})} = 1\} = \\ &= \Pr\{1 - \varphi(1_i, \mathbf{x}) = 0\} \cdot \Pr\{1 - \overline{\varphi(0_i, \mathbf{x})} = 0\} = \Pr\{1 - \varphi(1_i, \mathbf{x}) = 0\} \cdot \Pr\{1 - (1 - \varphi(0_i, \mathbf{x})) = 0\} = \\ &= \Pr\{\varphi(0_i, \mathbf{x}) = 0\} - \Pr\{\varphi(1_i, \mathbf{x}) = 0\} \cdot \Pr\{\varphi(0_i, \mathbf{x}) = 0\} \end{aligned}$$

and for a coherent system with a monotonically structure function [29]:

$$\begin{aligned} I_B(x_i) &= \Pr\{\varphi(0_i, \mathbf{x}) = 0\} - \Pr\{\varphi(1_i, \mathbf{x}) = 0\} \cdot \Pr\{\varphi(0_i, \mathbf{x}) = 0\} = \\ &= \Pr\{\varphi(0_i, \mathbf{x}) = 0\} - \Pr\{\varphi(1_i, \mathbf{x}) = 0 \wedge \varphi(0_i, \mathbf{x}) = 0\} = \Pr\{\varphi(0_i, \mathbf{x}) = 0\} - \Pr\{\varphi(1_i, \mathbf{x}) = 0\}. \end{aligned}$$

For example, the structure function of the system (Table 1) has 5 nonzero values for the derivatives $\partial\phi(\mathbf{x})/\partial x_1$, so the first component BI is calculated as:

$$IB_1 = \Pr\{\partial\phi(\mathbf{x})/\partial x_1 = 1\} = q_2p_3p_4 + p_2q_3q_4 + p_2q_3p_4 + p_2p_3q_4 + p_2p_3p_4,$$

and for other components:

$$IB_2 = \Pr\{\partial\phi(\mathbf{x})/\partial x_2 = 1\} = p_1q_3q_4 + p_1q_3p_4 + p_1p_3q_4.$$

$$IB_3 = \Pr\{\partial\phi(\mathbf{x})/\partial x_3 = 1\} = p_1q_2p_4$$

$$IB_4 = \Pr\{\partial\phi(\mathbf{x})/\partial x_4 = 1\} = p_1q_2p_3.$$

Therefore SI, MSI and BI indices can be calculated based on Direct Partial Boolean Derivatives.

Criticality Importance (CI). BI (20) describes the influence of a failure of the i -th system component on the system's availability, but doesn't take into account the probability of this component's failure. CI adjusts it and is defined as the probability that the i -th system component is relevant to the system's failure if it has failed [19]:

$$CI_i = BI_i \cdot \frac{q_i}{U}. \tag{21}$$

where BI_i is the i -th system component BI measure (20); q_i is probability of the i -th system failure (1) and U is the system's unavailability (4).

Dynamic Reliability Indices (DRI). DRI have been considered in paper [36]. DRIs allow the estimation of a component relevant to system failure. There are two groups of DRI: *Component Dynamic Reliability Indices (CDRI)* and *Dynamic Integrated Reliability Indices (DIRI)*.

CDRI indicates the influence of the i -th component's fault on the system's failure and is similar to the definition of modified SI, but CDRI includes two probabilities: (a) the probability of the system's failure caused by the i -th component's inoperation and (b) the probability of a component failure:

$$CDRI_i = MSI_i \cdot q_i. \tag{22}$$

where MSI_i is defined by (19); q_i is probability of a component failure (1).

DIRI is the probability of system failure that is caused by one of the system components in-operation (one of n):

$$DIRI = \sum_{i=1}^n ICDRI_i \prod_{\substack{q=1 \\ q \neq i}}^n (1 - ICDRI_q). \tag{23}$$

Therefore SI, MSI, BI, CI, CDRI and DIRI indices can be calculated based on Direct Partial Boolean Derivatives and the development of the algorithm for computation of these derivatives is important step in importance analysis based on quantification estimation of IMs.

3.2. Direct Partial Boolean Derivative Calculation based on BDD

One of possible BDD-based method for calculating IMs has been presented in [14]. The authors of the paper [14] proposed to define components states by the structure function BDD for computation of IM according to typical equation of IMs. Logical Differential Calculus has been used for calculation of some IMs in [35]. However, the proposed algo-

gorithms don't account for specific of the system with two states (available and unavailable).

We develop two algorithms for calculating Direct Partial Boolean Derivative (11) based on BDD of the structure function. These algorithms have identical basic principle: it is construction of the "tree of paths" by analyzing the paths from the root to the sink node of the BDD that agrees with conditions:

$$\varphi(0_i, \mathbf{x}) = 0 \quad \text{and} \quad \varphi(1_i, \mathbf{x}) = 1 \tag{24}$$

Therefore the tree of paths for condition $\varphi(0_i, \mathbf{x}) = 0$ unites all paths from the root to the sink node 0 that include out coming edges of the non-sink node x_i labelled 0. The tree of paths for the condition $\varphi(1_i, \mathbf{x}) = 1$ is constructed similarly.

For example, consider the construction of the tree of paths for the series-parallel system by the BDD (Fig. 4). Form the tree of paths of the derivative $\partial\phi(\mathbf{x})/\partial x_2$ for the condition $\varphi(0_2, \mathbf{x}) = 0$ (Fig. 6). The variable, on which the derivative is calculated, isn't included in the

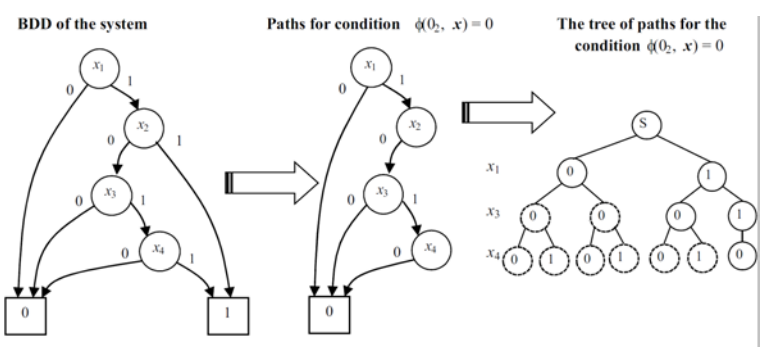


Fig. 6. The tree of paths for condition of the BDD in Fig. 4

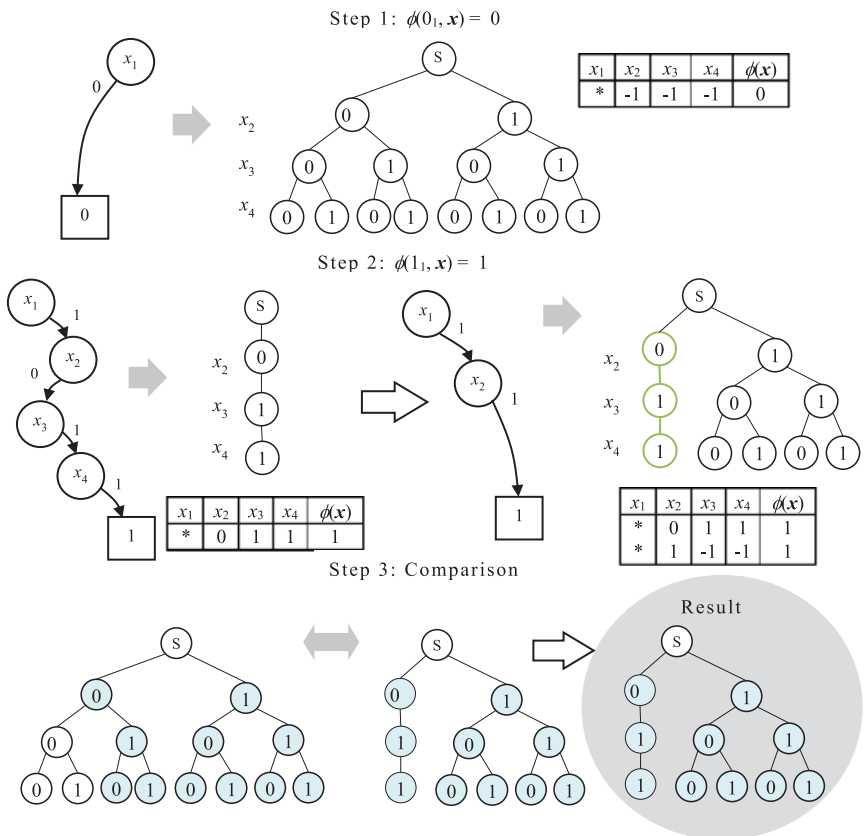


Fig. 7. Example of the Algorithm 1

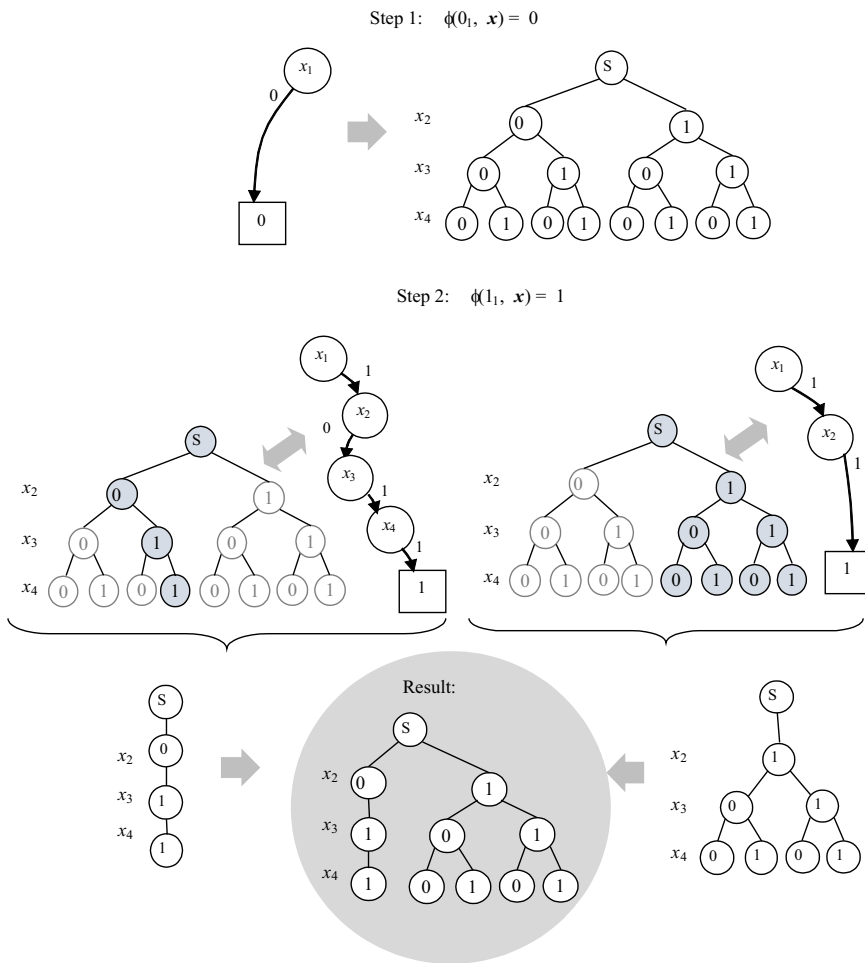


Fig. 8. Example of the Algorithm 2

tree of paths and conforms to the root labelled "S". It is the second variable x_2 in this example. Two paths are analysed for the specified condition (Fig. 6). The first (left) path has value of the first variable as 0 ($x_1 = 0$) and doesn't include the variables x_3 and x_4 . Therefore these variables (x_3 and x_4) can have any value (these values are indicated in the tree of paths by dotted line). The second path has variable $x_1 = 1$ and variable x_3 with two values. In case of $x_3 = 0$ the variable x_4 can have any value because it is absent in this path. But the variable $x_4 = 0$ if $x_3 = 1$. Therefore the tree of paths in Fig. 6 locates all paths for the condition $\phi(0_2, x) = 0$.

Two algorithms are developed for the Direct Partial Boolean Derivative (11) calculation by the BDD based on the use of the trees of paths.

The Algorithm 1 has three steps. The tree of paths for the condition $\phi(0_i, x) = 0$ is formed at the first step. The tree of paths for the condition $\phi(1_i, x) = 1$ is constructed at the second step. The third step of the algorithm compares these two trees. The general part of these trees is a decision that is corresponded to non-zero values of the Direct Boolean Derivative (11).

For example, Fig. 7 illustrates the calculation of the Direct Partial Boolean Derivative $\partial\phi(x)/\partial x_1$ based on the Algorithm 1 for BDD of the structure function (6) of the series-parallel system in Fig.1 with the structure function (6). The BDD of this function is presented in Fig.4. The first step of the algorithm is forming the tree of paths for condition $\phi(0_1, x) = 0$. This tree building is shown in detail for the second variable $\phi(0_2, x) = 0$ above (Fig. 6). The tree for condition $\phi(0_1, x) = 0$ includes all possible values of variables x_2, x_3 and x_4 , because the subdiagram for this condition doesn't include nodes of these variables. The second step of the algorithm permits to obtain the tree of paths for condition $\phi(1_1, x) = 1$. This tree includes all paths from the outcoming edge labelled 1 of the first variable x_1 to the sink node 1 of the BDD. The third step of the algorithm is comparing two trees that satisfy condition $\phi(0_1, x) = 0$ and $\phi(1_1, x) = 1$ accordingly. The final tree of paths (the algorithm result) includes paths that are equal for two trees. Therefore, the final tree includes 5 paths that are conformed with non-zero values of the derivative $\partial\phi(x)/\partial x_1: x = (1 \leftrightarrow 0, 0, 1, 1), x = (1 \leftrightarrow 0, 1, 0, 0), x = (1 \leftrightarrow 0, 1, 0, 1), x = (1 \leftrightarrow 0, 1, 1, 0), x = (1 \leftrightarrow 0, 1, 1, 1)$.

Algorithm 1 and Algorithm 2 are tested based on the sets of benchmarks LGSynth91 with the tool ABC (A System for Sequential Synthesis and Verification developed by Berkeley Verification and Synthesis Research Center) [20]. This benchmark has the PLA – EXPRESSO format and Table 3 summarizes the selected benchmarks, specifying the number of input variables, the number of output functions, the number of product terms appearing in the benchmark and their analysis for BDDs construction.

Table 3. Benchmark characteristics and analysis for BDD

Benchmark	Input	Output	Product terms	Number of non-terminal nodes in the BDD	Number of variables of the structure function
5xp1	7	10	75	5	14
alu4	14	8	1028	11	163
apex1	45	45	206	9	15
apex3	54	50	280	9	46
apex4	9	19	438	10	136
b12	15	9	431	5	8
bw	5	28	87	6	20
clip	9	5	167	10	68
con1	7	2	9	7	13
cps	24	109	654	19	105
duke2	22	29	87	8	41
e64	65	65	65	10	10
ex1010	10	10	1024	11	459
ex4	128	28	620	10	76
ex5	8	63	256	8	8
inc	7	9	34	7	15
misex1	8	7	32	7	12
misex2	25	18	29	13	13
misex3	14	14	1848	15	1640
misex3c	14	14	305	9	103
pdC	16	40	2810	15	21
rd53	5	3	32	6	15
rd73	7	3	141	8	70
rd84	8	4	256	9	130
sao2	10	4	58	11	76
seq	41	35	1459	13	18
squar5	5	8	32	6	6
Z5xp1	7	10	128	8	8

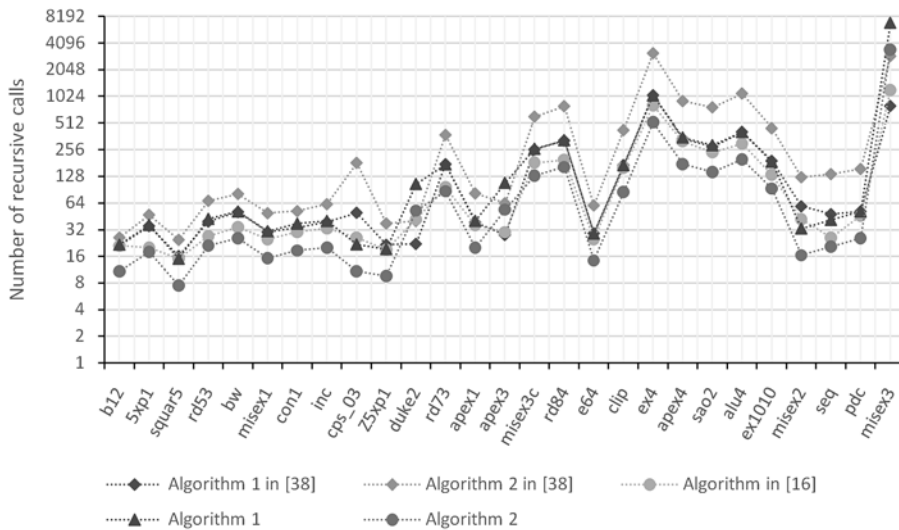


Fig. 9. Comparison of algorithms computational complexity by number of recursive calls

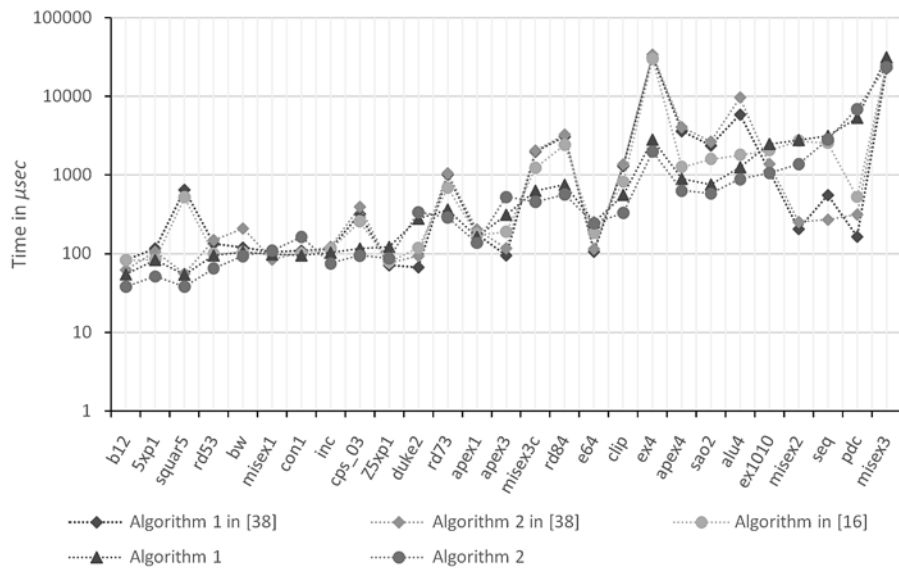


Fig. 10. Comparison of algorithms computational complexity by computational time (in usec)calls

Table 4. The calculation of indexes SI , SIM , BI , CI , $CDRI$ for benchmark $rd84$

x_i	q_i	p_i	SI_i [3]	BI_i [6]	CI_i [12]	New algorithm		
						SI_i	BI_i	CI_i
1	0.3	0.47	0.0078125	0.122472	0.0401868	0.0078125	0.122472	0.0401868
2	0.1	0.9	0.0078125	0.095256	0.0104188	0.0078125	0.095256	0.0104188
3	0.4	0.6	0.0078125	0.142884	0.0625129	0.0078125	0.142884	0.0625129
4	0.5	0.5	0.0078125	0.171461	0.0937693	0.0078125	0.171461	0.0937693
5	0.1	0.9	0.0078125	0.095256	0.0104188	0.0078125	0.095256	0.0104188
6	0.3	0.7	0.0078125	0.122472	0.0401868	0.0078125	0.122472	0.0401868
7	0.2	0.8	0.0078125	0.107163	0.0234423	0.0078125	0.107163	0.0234423
8	0.1	0.9	0.0078125	0.095256	0.0104188	0.0078125	0.095256	0.0104188

These benchmarks are used for the comparison, control and examination of two algorithms. In addition these algorithms have been compared with the similar algorithm that has been proposed in [16] and with algorithm proposed in [38]. Experiments permit the computational complexity of the proposed algorithms to be estimated. Computational complexities such as the number of recursive calls of algorithms are shown in Fig. 9 and the time for computation of non-zero values of Direct Partial Boolean Derivatives is presented in Fig. 10.

The analysis of the data in Fig.9 and Fig.10 shows that the two proposed algorithms have similar characteristics but Algorithm 2 has lesser computational complexity, which is well seen in the chart on Fig. 9. Characteristics of the algorithms in [16] and [38] are worse in comparison of proposed algorithms. Therefore, the Algorithm 2 can be used for the calculation of non-zero values of Direct Partial Boolean Derivatives for the structure function preferable. The non-zero values of Direct Partial Boolean Derivatives permit to calculate IMs (18) – (23). It is important that these equations are agree with well know definitions of IMs. Therefore IMs values calculated by the proposed Algorithm 1 and Algorithm 2 based on the definitions (18) – (23) are equal IMs values computed by the definition from [3, 6, 12].

For example, consider the benchmark $rd84$ (Table 3). In the Table 4 SI , BI and CI values are shown that are calculated by new algorithm (Algorithm 2) and traditional algorithm. This Table consists of two parts. The first of them include IMs values that are calculated based on the definitions in papers [3, 6, 12]. IMs values computed by new algorithms are described in the second part of this Table. Note IMs values are equal for two proposed algorithms (Algorithm 1 and Algorithm 2). Therefore IMs values for two different approaches are equal, because new algorithms realize equations for IMs calculation (18), (19) and (21) – (23) that have the mathematical substantiation of the correlation with IMs definitions in [3], [6], [12].

4. Conclusions and Future Work

This work presented an efficient approach to analyze the reliability and importance analysis. This approach can be summarized in following characteristics. First characteristic of proposed approach is possibility of calculating IMs using mathematical apparatus Direct Partial Boolean Derivatives, which is based on new equations for IMs (18) – (23). The background of this approach is the computation of non-zero values of derivatives that are used in these equations of IMs. Next aspect is based on development of new algorithms for calculation of Direct Partial Boolean Derivatives based on BDD, which allows analysis of the function of large dimensions. The last important point of presented work is the introduction an experimental analysis based on BDD algorithms, which are highly efficiently both in computational time and storage demand for importance analysis and they also make it possible for us to study practical and large systems. Research based on this approach as sensitivity analysis, importance measures, and optimal design issues of large systems will become more important in the future. Algorithms proposed in this work can be generalized for the non-coherent system. In this case the calculation of IMs will be based on the analysis of two Direct Partial Boolean Derivatives $\partial\varphi(0 \rightarrow 1)/\partial x_i(0 \rightarrow 1)$ and $\partial\varphi(0 \rightarrow 1)/\partial x_i(1 \rightarrow 0)$

. Proposed algorithms can be modified for the integration to IMs [13, 19], which could allow investigating of influence of some system components to reliability or availability of system. These measures will be defined based on the Direct Partial Boolean Derivative with respect to variable vector [36]. All these generalization (for non-co-

herent system and integrated IMs) can be implemented based on non-important modification of Direct Partial Logic Derivatives. The derivative with respect to variable vector will be exploited for the integrated IMs calculation.

Acknowledgement

This work was partially supported by grant of Scientific Grant Agency of the Ministry of Education of Slovak Republic (Vega 1/0498/14).

References

1. Akers SB. Binary Decision Diagrams. IEEE Transaction on Computers 1978; 27: 509–16, <http://dx.doi.org/10.1109/TC.1978.1675141>.
2. Akers SB. On a Theory of Boolean Functions. Journal of the Society for Industrial and Applied Mathematics 1959; 7: 487–98, <http://dx.doi.org/10.1137/0107041>.
3. Armstrong MJ. Reliability-importance and dual failure-mode components. IEEE Transaction on Reliability 1997; 46: 212–21, <http://dx.doi.org/10.1109/24.589949>.
4. Barlow RE, Proschan F. Importance of system components and fault tree events. Stochastic Processes and their Applications 1975; 3: 153–73, [http://dx.doi.org/10.1016/0304-4149\(75\)90013-7](http://dx.doi.org/10.1016/0304-4149(75)90013-7).
5. Beeson S, Andrews JD. Importance measures for noncoherent-system analysis. IEEE Transactions on Reliability 2003; 52: 301–10, <http://dx.doi.org/10.1109/TR.2003.816397>.
6. Birnbaum ZW. On importance of difference components in a multi-component system. Multi-Variant Anal 2 1969: 581–92.
7. Bochmann D, Posthoff C. Binäre dynamische Systeme. Berlin: Akademie-Verlag; 1981.
8. Borgonovo E. The reliability importance of components and prime implicants in coherent and non-coherent systems including total-order interactions. European Journal of Operational Research 2010; 204: 485–95, <http://dx.doi.org/10.1016/j.ejor.2009.10.021>.
9. Bryant RE. Graph-Based Algorithms for Boolean Function Manipulation. IEEE Transactions on Computers 1986; C-35: 677–91, <http://dx.doi.org/10.1109/TC.1986.1676819>.
10. Davio M., J.-P. Deschamps, A. Thayse, Discrete and switching functions. St. Saphorin, Switzerland: Georgi Pub. Co.; New York: McGraw-Hill International Book Co; 1978.
11. Dufлот N, Bérenguer C, Dieulle L, Vasseur D. A min cut-set-wise truncation procedure for importance measures computation in probabilistic safety assessment. Reliability Engineering & System Safety 2009; 94: 1827–37, <http://dx.doi.org/10.1016/j.res.2009.05.015>.
13. Gao X, Cui L, Li J. Analysis for joint importance of components in a coherent system. European Journal of Operational Research 2007; 182: 282–99, <http://dx.doi.org/10.1016/j.ejor.2006.07.022>.
14. Chang Y-R, Amari S V, Kuo S-Y. Computing system failure frequencies and reliability importance measures using OBDD. IEEE Transaction on Computers 2004; 53: 54–68, <http://dx.doi.org/10.1109/TC.2004.1255790>.
15. Chang Y-RY, Huang CY, Kuo S. Performance assessment and reliability analysis of dependable and distributed computing systems based on BDD and recursive merge. Applied Mathematics and Computation 2010; 217: 403–13, <http://dx.doi.org/10.1016/j.amc.2010.05.075>.
16. Changqian W, Chenghua W. A Method for Logic Circuit Test Generation Based on Boolean Partial Derivative and BDD. 2009 WRI World Congress on Computer Science and Information Engineering 2009; 3: 499–504, <http://dx.doi.org/10.1109/CSIE.2009.44>.
17. Jung WS, Han SH, Ha J. A fast BDD algorithm for large coherent fault trees analysis. Reliability Engineering & System Safety 2003; 83 (3): 369–374, <http://dx.doi.org/10.1016/j.res.2003.10.009>.
18. Kim Bjorkman, Solving dynamic flowgraph methodology models using binary decision diagrams, Reliability Engineering & System Safety, 2013; 111: 206–216, <http://dx.doi.org/10.1016/j.res.2012.11.009>.
19. Kuo W, Zhu X. Importance Measures in Reliability, Risk, and Optimization. Chichester, UK: John Wiley & Sons, Ltd; 2012, <http://dx.doi.org/10.1002/9781118314593>.
20. Laboratory CB and EA. The Benchmark Archives at CBL 2013.
21. Liudong Xing, Akhilesh Shrestha, Yuanshun Dai, Exact combinatorial reliability analysis of dynamic systems with sequence-dependent failures, Reliability Engineering & System Safety 2011; 96 (10): 1375–1385, <http://dx.doi.org/10.1016/j.res.2011.05.007>.
22. Liudong Xing, An Efficient Binary-Decision-Diagram-Based Approach for Network Reliability and Sensitivity Analysis, IEEE Transactions on System, Man and Cybernetics – Part A: System and Humans 2008; 38(1).
23. Liudong Xing, Gregory Levitin, BDD-based reliability evaluation of phased-mission systems with internal/external common-cause Reliability Engineering & System Safety 2013; 112:145–153, <http://dx.doi.org/10.1016/j.res.2012.12.003>.
24. Moret BME, Thomason MG. Boolean Difference Techniques for Time-Sequence and Common-Cause Analysis of Fault-Trees. IEEE Transaction on Reliability 1984; R-33: 399–405, <http://dx.doi.org/10.1109/TR.1984.5221879>.
25. Nokland TE, Aven T. On selection of importance measures in risk and reliability analysis. International Journal of Performability Engineering 2013; 9: 133–47.
26. Posthoff C, Steinbach B. Logic Functions and Equations. Binary Models for Computer Science 2004, <http://dx.doi.org/10.1007/978-1-4020-2938-7>.
27. Rausand M, Hyland A, editors. System Reliability Theory. Hoboken, NJ, USA: John Wiley & Sons, Inc.; 1994, <http://dx.doi.org/10.1002/9780470316900>.
28. Rauzy A. Mathematical foundations of minimal cut sets. IEEE Transactions on Reliability 2001; 50: 389–96, <http://dx.doi.org/10.1109/24.983400>.
29. Ryabinin, IA. and Parfenov, YuM., Determination of "Weight" and "Importance" of Individual Elements at Reliability Estimation of a Complex System. Energetics Transport 1978, 6: 22–32.
30. Shmerko V, Lyshevski S, Yanushkevich S. Computer Arithmetics for Nanoelectronics. CRC Press 2009, <http://dx.doi.org/10.1201/b15950>.

31. Shumin Li, Shubin Si, Hongyan Dui, Zhiqiang Cai, Shudong Sun, A novel decision diagrams extension method, *Reliability Engineering & System Safety* 2014; 126: 107-115, <http://dx.doi.org/10.1016/j.ress.2014.01.017>.
32. Schneeweiss WG. A short Boolean derivation of mean failure frequency for any (also non-coherent) system. *Reliability Engineering & System Safety* 2009; 94: 1363–7, <http://dx.doi.org/10.1016/j.ress.2008.12.001>.
33. Talansev A. Analysis and synthesis of some electrical circuits by means of special logical operators. *Automation and Remote Control* 1959; 20; 898–907.
34. Tucker JH, Tapia MA, Bennett AW. Boolean Integral Calculus for Digital Systems. *IEEE Transactions on Computers* 1985; C-34; 78–81, <http://dx.doi.org/10.1109/TC.1985.1676517>.
35. Zaitseva E, Levashenko V. Multiple-Valued Logic mathematical approaches for multi-state system reliability analysis. *Journal of Applied Logic* 2013; 11: 350-362, <http://dx.doi.org/10.1016/j.jal.2013.05.005>.
36. Zaitseva E. Importance Analysis of a Multi-State System Based on Multiple-Valued Logic Methods. *Recent Advances in System Reliability - Signatures, Multi-State System, Statistical Inference - Springer* 2012; 113–34.
37. Zaitseva E., Levashenko V. Importance analysis by logical differential calculus. *Automation and Remote Control - Springer* 2013; 74: 171–82, <http://dx.doi.org/10.1134/S000511791302001X>.
38. Zaitseva E., Levashenko V., Kostolny J. Multi-State System Importance Analysis based on Direct Partial Logic Derivative. *International Conference on Quality, Reliability, Risk, Maintenance and Safety Engineering* 2012; 1514 – 1519, <http://dx.doi.org/10.1109/ICQR2MSE.2012.6246513>.

Elena ZAITSEVA
Vitaly LEVASHENKO
Jozef KOSTOLNY

Department of Informatics
University of Zilina, Faculty of Management Science and Informatics
Univerzitna 8215/1, 010 26 Zilina, Slovakia.
E-mail: elena.zaitseva@fri.uniza.sk, jozef.kostolny@fri.uniza.sk
