Reliability Analysis of Multi-State Systems based on Tools of Multiple-Valued Logic

Miroslav Kvassay, Elena Zaitseva, Jozef Kostolny, Vitaly Levashenko
Faculty of Management Science and Informatics
University of Zilina
Zilina, Slovakia
{miroslav.kvassay, elena.zaitseva, jozef.kostolny, vitaly.levashenko}@fri.uniza.sk

***Abstract*—Multi-State Systems (MSSs) are one of the basic mathematical models that are used in reliability analysis of complex systems. Analysis of such systems is a challenging task. It involves several steps from which one of the most important is investigation of influence of system components on system proper work. This investigation can be qualitative or quantitative. The qualitative analysis focuses on identification of conditions in which component degradation results in system deterioration. The quantitative analysis estimates probabilities of occurrences of such situations. In this paper, we propose a method for qualitative and quantitative analysis of topological properties of MSSs. The method is based on logical differential calculus and is implemented using multi-valued decision diagrams.**

***Keywords*—logical differential calculus; multi-valued decision diagram; reliability; multi-state system; structural importance**

I. Introduction

One of the principal steps of reliability engineering is creation of mathematical model of a considered system. As a rule, two different approaches are used in reliability analysis: Binary-State Systems (BSSs) [1] and Multi-State Systems (MSSs) [2, 3]. A BSS can be only in one of two possible states – functioning (represented by number 1) and failed (presented as number 0). These models are useful for the analysis of systems in which any deviation from perfect functioning can result in a disaster, e.g. nuclear power plants [4] or aviation systems [5]. However, there exist many systems that can also perform their mission in situations when they are not perfectly working. Typical examples of such systems are distribution networks or service systems [6, 7]. It can be quite complicated to model these systems as BSSs because a boundary between situations in which the system is considered failed and when it is functioning have to be known. However, this can be a problem. Therefore, MSSs are more appropriate for modeling such systems since they allow defining several, i.e. m, levels of system performance from perfectly functioning (represented by number m -1) to completely failed (presented as number 0).

Every system consists of one or more components. In BSSs, there is assumption that every system component can also be either failed or functioning. In the case of a MSS, system components can be in one of more than two states. The relationship that defines dependency between level/state of system performance and states of system components is known as the structure function $\phi(\boldsymbol{x})$ [2, 3, 7]:

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

where n is a number of system components, m is a number of system states (performance levels), m_i is a count of states of the i-th system component, and $\boldsymbol{x} = (x_1, x_2,\ldots, x_n)$ is a vector of components states (state vector). Specially, if $m_1 = m_2 = \ldots = m_n = m$, then the system is recognized as homogeneous [2, 3]. Please note that $m_1 = m_2 = \ldots = m_n = m = 2$ for BSSs.

Based on the properties of the structure function, two different classes of systems can be recognized: coherent and noncoherent. In what follows, we will assume that the system is coherent, which means that (a) every component is relevant, i.e. each component has influence on system state, and (b) the structure function is non-decreasing, i.e. there exists no situation in which component failure (degradation) results in system repair (improvement) [2, 3]. Furthermore, we will also assume that all components of a coherent MSS can degrade gradually state by state, i.e. if a component is in state s ($s > 0$), then it can degrade only to state s -1.

The structure function of a BSS can be interpreted as a Boolean function. This fact permits using some tools related to Boolean functions, such as Binary Decision Diagrams and Boolean derivatives, in reliability analysis of BSSs [8]. Similarly, the structure function of a homogenous MSS can be viewed as a Multiple-Valued Logic (MVL) function. This implies that some methods of MVL can be used in the analysis of MSSs [7, 9–11]. In this paper, we will focus on two of them – Multi-valued Decision Diagrams (MDDs) and Direct Partial Logic Derivatives (DPLDs).

II. Reliability Analysis

A. Availability

The structure function is very important in reliability analysis because it allows us to analyze topological properties of the system. However, it does not permit investigating other system characteristics, such as system availability, mean time to failure, or mean time to repair. For this purpose, probabilities of individual states of system components have to be known:

$$p_{i,s} = \Pr\{x_i = s\}, \; s = 0,1,\ldots, m_i -1. \quad (2)$$

This work was supported by the grant of 7th RTD Framework Program No 610425 (RASimAs) and VEGA grant 1/0498/14.

978-1-4799-8569-2/15/$31.00

Knowledge of these probabilities and the structure function allows us to compute the probability that the system is in state j. This knowledge can also be used to calculate availability and unavailability of the considered system, which are defined with respect to system state j as follows [2, 3]:

$$A^{\geq j} = \Pr\{\phi(\boldsymbol{x}) \geq j\},\ U^{\geq j} = \Pr\{\phi(\boldsymbol{x}) < j\}, \quad j = 1,2,\ldots,m-1. \tag{3}$$

Clearly, availability for system state j agrees with the probability that the system is in such state that its performance can satisfy a demand corresponding to state j.

B. Performance Utility Function

System states are abstract numbers. They can be joined with a physical meaning using the concept of utilities attached to them [12]. (For example, in the case of a power supply system, a utility attached to state j can be output performance measured in MW.) If we assume that utility o_j is attached to system state j, then the expected utility of the considered MSS is known as performance utility function, and it is defined as follows [12]:

$$O = \sum_{j=0}^{m-1} o_j \Pr\{\phi(\boldsymbol{x}) = j\} = o_0 + \sum_{j=1}^{m-1} \left(o_j - o_{j-1}\right) A^{\geq j}. \tag{4}$$

Specially, if $o_j = j$ for all $j \in \{0,1,\ldots,m-1\}$, then this function estimates the expected state of the system.

C. Component Criticality

Criticality is one of the key concepts of reliability analysis. It is used in the qualitative analysis to describe situations in which a minor degradation (degradation by one state) of a system component results in system deterioration. Depending on whether we focus on a system state or on availability of the system, two different approaches can be recognized.

The first approach focusing on a system state can be found in [13]. According to this work, state s of component i is critical for degradation of system state j at state vector $(\cdot_i, \boldsymbol{x})$ if $\phi(s_i, \boldsymbol{x}) = j$ and $\phi((s-1)_i, \boldsymbol{x}) < j$. (Please note that $(\cdot_i, \boldsymbol{x})$ denotes state vector $\boldsymbol{x}$ in which the i-th element is not presented, i.e. $(\cdot_i, \boldsymbol{x}) = (x_1, x_2,\ldots, x_{i-1}, x_{i+1},\ldots, x_n)$ and, in the similar way, $(s_i, \boldsymbol{x}) = (x_1, x_2,\ldots, x_{i-1}, s_i, x_{i+1},\ldots, x_n)$.) Clearly, criticality of state s of component i for system state j implies that the i-th component makes the difference between situations in which the system is in state j and in which it is degraded.

Another approach focuses on system availability, and it can be recognized in several works, e.g. [2, 12, 14]. Based on these works, we say that state s of component i is critical for degradation of system availability calculated for state j at state vector $(\cdot_i, \boldsymbol{x})$ if $\phi(s_i, \boldsymbol{x}) \geq j$ and $\phi((s-1)_i, \boldsymbol{x}) < j$. So, this approach is used to describe circumstances in which a state of component i makes the difference between situations in which the system is available and when it is unavailable (with respect to system state j).

State vectors at which a state of a system component is critical for degradation of system state or system availability are known as critical path vectors for the component state. These state vectors are very important in quantification of influence of component degradation on system activity because one of the typical approaches used in such analysis is based on the assumption that a component whose states are critical in more situations, i.e. a component for whose states more critical path vectors exist, is more important than a component whose states are critical in fewer situations.

D. Structural Importance

The availability is an important characteristic since it can be used to estimate time during which the system will be available. However, it does not allow analyzing influence of individual system components on system activity. For this purpose, other measures that are known as Importance Measures (IMs) are used. The most commonly known are the Structural Importance (SI), Birnbaum's Importance (BI), and the Criticality Importance (CI) (Table I). In what follows, we will mainly focus on the SI.

TABLE I. IMPORTANCE MEASURES

Importance Measures	Meaning
SI	The SI concentrates on the topological structure of the system, and it corresponds to the relative number of situations in which a component (state) is critical for system degradation.
BI	The BI of a given component is defined as the probability that the component (state) is critical for system degradation.
CI	The CI of a given component is calculated as the probability that the system deterioration has been caused by the component degradation given that the system is not perfectly functioning.

The SI concentrates on topological properties of the system. It has originally been developed for the analysis of BSSs [15]. Its use in reliability analysis of MSSs has been considered in several works. In [14], the SI has been defined as follows:

$$\mathrm{SI}_{i,s}^{\geq j} = \frac{\left|\{(\cdot_i, \boldsymbol{x}) : \phi(s_i, \boldsymbol{x}) \geq j \text{ and } \phi((s-1)_i, \boldsymbol{x}) < j\}\right|}{\prod_{l=1, l \neq i}^{n} m_l}, \tag{5}$$

where $\{(\cdot_i, \boldsymbol{x}) : \phi(s_i, \boldsymbol{x}) \geq j \text{ and } \phi((s-1)_i, \boldsymbol{x}) < j\}$ is a set of all critical path vectors for system availability computed with respect to state j and for state s of component i, $|.|$ denotes size (number of elements) of a set, n is a number of system components, m_l for $l = 1,2,\ldots, i-1, i+1,\ldots, n$ agrees with a count of states of component l, and $\prod_{l=1, l \neq i}^{n} m_l$ is a number of all state vectors in which the i-th system component is in state s. Clearly, this SI recognizes a relative number of situations in which degradation of state s of component i causes degradation of system availability computed for state j of the system.

In the similar way, author of paper [13] has proposed the next definition of this measure:

$$\mathrm{SI}_{i,s}^{j\downarrow} = \frac{\left|\{(\cdot_i, \boldsymbol{x}) : \phi(s_i, \boldsymbol{x}) = j \text{ and } \phi((s-1)_i, \boldsymbol{x}) < j\}\right|}{\prod_{l=1, l \neq i}^{n} m_l}. \quad (6)$$

According to this formula, the SI identifies a relative number of situations in which a minor degradation of state s of component i results in degradation of system state j, i.e. this measure is defined as a relative count of critical path vectors for system state j and for state s of component i.

Another approach has been considered in [12]. Based on this paper, the SI can be formulated in such way that it will analyze importance of a given component state on the entire system (not only on a specified system state or availability):

$$\mathrm{SI}_{i,s}^{\geq} = \frac{\sum_{j=1}^{m-1} \left(o_j - o_{j-1}\right) \left|\{(\cdot_i, \boldsymbol{x}) : \phi(s_i, \boldsymbol{x}) \geq j \text{ and } \phi((s-1)_i, \boldsymbol{x}) < j\}\right|}{\prod_{l=1, l \neq i}^{n} m_l}. \quad (7)$$

Finally, the following definition of the SI has been used in [13] to investigate the total importance of a given component on system activity:

$$\mathrm{SI}_{i}^{\downarrow} = \frac{\sum_{s=1}^{m_i-1} \sum_{j=1}^{m-1} \sum_{k=1}^{j} \left(o_j - o_{j-k}\right) \left|\{(\cdot_i, \boldsymbol{x}) : \phi(s_i, \boldsymbol{x}) = j \text{ and } \phi((s-1)_i, \boldsymbol{x}) = j-k\}\right|}{\prod_{l=1, l \neq i}^{n} m_l}, \quad (8)$$

where $\{(\cdot_i, \boldsymbol{x}) : \phi(s_i, \boldsymbol{x}) = j \text{ and } \phi((s-1)_i, \boldsymbol{x}) = j-k\}$ is a set of all state vectors at which degradation of component i from state s to s -1 results in degradation of the system from state j to j -k. If system states are join with utilities o_j, then this degradation results in decrease of o_j -o_{j-k} in the value of the system utility. Since there are $\prod_{l=1, l \neq i}^{n} m_l$ different state vectors at which the i-th component state can degrade from value s to s -1, expression $\left|\{(\cdot_i, \boldsymbol{x}) : \phi(s_i, \boldsymbol{x}) = j \text{ and } \phi((s-1)_i, \boldsymbol{x}) = j-k\}\right| / \prod_{l=1, l \neq i}^{n} m_l$ agrees with the relative number of situations in which a minor degradation of state s of component i results in degradation of the system from state j to j -k. If we multiply this expression by difference o_j -o_{j-k}, then we can recognize the consequences of the considered system degradation on the system utility caused by a minor degradation of state s of component i. Next, if we do summation of these expressions for all possible values of k, i.e. 1,2,…,j, then we get the topological influence of a minor degradation of the considered component state on the system utility when the system is in state j. Doing other summations for all possible j and s, we can obtain the total topological influence of the considered component on the system activity defined by the utilities attached to individual system states.

Definitions (5) – (8) show several ways for preforming importance analysis in MSSs, i.e. we can identify components that have the greatest influence on the whole system, or we can find components that are the most important for a given system state (availability) or component states with the greatest influence on the system or, finally, we can investigate the dependency between degradation of a given component state and a given system state (availability).

III. Logical Differential Calculus

A. Direct Partial Logic Derivatives

Logical differential calculus is a special tool that has been developed for analysis of dynamic properties of logic functions. Direct Partial Logic Derivatives (DPLDs) are part of this tool. They are used to identify situations in which change of a logic variable from value s to r results in a change of logic function from value j to h. For MVL functions, this derivative is defined as follows [16]:

$$\frac{\partial\phi(j \to h)}{\partial x_i(s \to r)} = \begin{cases} 1, & \text{if } \phi(s_i, \boldsymbol{x}) = j \text{ and } \phi(r_i, \boldsymbol{x}) = h \\ 0, & \text{otherwise} \end{cases}, \quad (9)$$
$$\text{for } s,r,j,h \in \{0,1,\ldots,m\text{-}1\}, s \neq r, j \neq h.$$

Use of DPLDs in reliability analysis of MSSs has been considered in several papers. Works [9, 10] have primarily focused on their use in the analysis of homogenous systems. It has been shown that they can be used to find situations in which a minor degradation of component state results in system degradation, i.e. to find state vectors at which state s of component i is critical for degradation of system state j. Other works have considered a modification of definition (9), which allows using DPLDs also in the analysis of non-homogeneous systems [7, 11]:

$$\frac{\partial\phi(j \to h)}{\partial x_i(s \to r)} = \begin{cases} 1, & \text{if } \phi(s_i, \boldsymbol{x}) = j \text{ and } \phi(r_i, \boldsymbol{x}) = h \\ 0, & \text{otherwise} \end{cases}, \quad (10)$$
$$\text{for } s,r \in \{0,1,\ldots,m_i\text{-}1\}, s \neq r;\ j,h \in \{0,1,\ldots,m\text{-}1\}, j \neq h.$$

This slight modification allows us to use DPLDs to find the coincidence between degradation of a system of any type and degradation of its components.

B. Integrated Direct Partial Logic Derivatives

DPLDs give us a detailed view on the dependency between component degradation and system degradation, but the main problem is that there exist a lot of them for one component, i.e. if we want to analyze consequences of a minor degradation of a given component state on system state j, then we have to compute j -1 various derivatives, i.e. all DPLDs of the form of $\partial\phi(j \to h)/\partial x_i(s \to s-1)$ in which $h < j$. Another fact is that DPLDs carry quite little information, i.e. they can contain very few nonzero elements because if a minor degradation of state s of the i-th system component causes decrease in system state from value j to h for state vector (s_i, $\boldsymbol{x}$), then all other DPLDs of the form of $\partial\phi(\bar{j} \to \bar{h})/\partial x_i(s \to s-1)$ where $\bar{j} \neq j$ or $\bar{h} \neq h$ have to take zero value for this state vector. These facts indicate that the original version of DPLDs might not be very suitable for investigation of MSSs in which a minor degradation of component state can cause degradation of the system by more than one state. Because of that, we propose new types of logic derivatives that will be named as Integrated Direct Partial Logic Derivatives (IDPLDs). This name reflects the fact that these derivatives are composed of several DPLDs.

Firstly, we can introduce IDPLDs identifying situations in which a change of component state results in a degradation of system state j:

$$\frac{\partial\phi(j\downarrow)}{\partial x_i(s \to r)} = \bigcup_{h=0}^{j-1} \frac{\partial\phi(j \to h)}{\partial x_i(s \to r)} = \begin{cases} 1, & \text{if } \phi(s_i, \boldsymbol{x}) = j \text{ and } \phi(r_i, \boldsymbol{x}) < j \\ 0, & \text{otherwise} \end{cases}, \quad \text{for } s, r \in \{0,1,\ldots,m_i\text{-}1\}, s \neq r,\ \ j \in \{1,2,\ldots,m\text{-}1\}, \tag{11}$$

or in falling the system into a given state:

$$\frac{\partial\phi(\downarrow j)}{\partial x_i(s \to r)} = \bigcup_{h=j+1}^{m-1} \frac{\partial\phi(h \to j)}{\partial x_i(s \to r)} = \begin{cases} 1, & \text{if } \phi(s_i, \boldsymbol{x}) > j \text{ and } \phi(r_i, \boldsymbol{x}) = j \\ 0, & \text{otherwise} \end{cases}, \quad \text{for } s, r \in \{0,1,\ldots,m_i\text{-}1\}, s \neq r,\ \ j \in \{0,1,\ldots,m\text{-}2\}, \tag{12}$$

where symbol $\bigcup$ denotes logical disjunction/union of DPLDs. Clearly, IDPLD (11) can be used to find state vectors $(._i, \boldsymbol{x})$ at which state s of component i is critical for degradation of system state j (i.e. its nonzero elements correspond to critical path vectors for state s of component i), while IDPLD (12) is useful for recognizing state vectors $(._i, \boldsymbol{x})$ at which state s of component i is critical for falling the system into state j.

IDPLDs (11) and (12) are useful in the analysis focusing on system state. Therefore, they could be used in computation of the SI measures (6) and (8). However, they are not suitable for identification of state vectors at which state s of component i is critical for degradation of system availability computed with regard to a given system state. For this purpose, the next IDPLDs, which have been defined in [7], are more appropriate:

$$\frac{\partial\phi(h_{\geq j} \to h_{<j})}{\partial x_i(s \to r)} = \bigcup_{h_u=j}^{m-1} \bigcup_{h_d=0}^{j-1} \frac{\partial\phi(h_u \to h_d)}{\partial x_i(s \to r)} = \begin{cases} 1, & \text{if } \phi(s_i, \boldsymbol{x}) \geq j \text{ and } \phi(r_i, \boldsymbol{x}) < j \\ 0, & \text{otherwise} \end{cases}, \quad \text{for } s, r \in \{0,1,\ldots,m_i\text{-}1\}, s \neq r,\ \ j \in \{1,2,\ldots,m\text{-}1\}. \tag{13}$$

If we assume that $r = s$ -1, then this integrated derivative detects all situations in which a minor degradation of state s of component i causes that the system changes from a state greater than or equal to j (notation $h_{\geq j}$) to a state less than j (notation $h_{<j}$).

The main difference between DPLDs and IDPLDs is that the former use only relation of equality while the latter admit using relations of greater/less than (or equal to). This facilitates identification of situations in which a given component state is critical for system degradation. For example, IDPLD (11) allows us to detect all situations in which state s of component i is critical for degradation of system state j. However, if we want to use DPLDs (10) for this task, then j -1 of them have to be calculated and their union has to be identified.

C. Computation of Structural Importance Measures using Integrated Direct Partial Logic Derivatives

Based on the correlation between critical state vectors and IDPLDs, the SI measures (5) – (7) can be computed as follows:

$$\begin{aligned} \mathrm{SI}_{i,s}^{\geq j} &= \mathrm{TD}(\partial\phi(h_{\geq j} \to h_{<j})/\partial x_i(s \to s-1)), \\ \mathrm{SI}_{i,s}^{j\downarrow} &= \mathrm{TD}(\partial\phi(j\downarrow)/\partial x_i(s \to s-1)), \\ \mathrm{SI}_{i,s}^{\geq} &= \sum_{j=1}^{m-1} (o_j - o_{j-1}) \mathrm{TD}(\partial\phi(h_{\geq j} \to h_{<j})/\partial x_i(s \to s-1)), \end{aligned} \tag{14}$$

where TD(.) denotes truth density of the argument interpreted as a function with Boolean-valued output, i.e. its value agrees with the relative number of points in which the function is nonzero.

The SI (8) can be expressed using DPLDs in the next way:

$$\mathrm{SI}_i^{\downarrow} = \sum_{s=1}^{m_i-1} \sum_{j=1}^{m-1} \sum_{k=1}^{j} (o_j - o_{j-k}) \mathrm{TD}(\partial\phi(j \to j-k)/\partial x_i(s \to s-1). \tag{15}$$

This formula can be rewritten in the following manner using IDPLDs (11) and (12):

$$\begin{aligned} \mathrm{SI}_i^{\downarrow} &= \sum_{s=1}^{m_i-1} \sum_{j=1}^{m-1} o_j \sum_{h=0}^{j-1} \mathrm{TD}(\partial\phi(j \to h)/\partial x_i(s \to s-1) \\ &\quad - \sum_{s=1}^{m_i-1} \sum_{h=0}^{m-1} o_h \sum_{j=h+1}^{m-1} \mathrm{TD}(\partial\phi(j \to h)/\partial x_i(s \to s-1) \\ &= \sum_{j=1}^{m-1} \sum_{s=1}^{m_i-1} o_j\ \mathrm{TD}(\partial\phi(j\downarrow)/\partial x_i(s \to s-1)) \\ &\quad - \sum_{j=0}^{m-2} \sum_{s=1}^{m_i-1} o_j\ \mathrm{TD}(\partial\phi(\downarrow j)/\partial x_i(s \to s-1)). \end{aligned} \tag{16}$$

This implies that all types of the SI measures investigating various dependencies between component degradation and deterioration of the system can be computed based on IDPLDs. The fact that we need to compute truth densities of IDPLDs implies that an efficient method for identification of nonzero elements of IDPLDs is required. For this purpose, we can use and customize the approach considered in [11] that is based on Multi-valued Decision Diagrams (MDDs).

IV. MULTI-VALUED DECISION DIAGRAMS

MDDs represent an efficient realization of MVL functions [17]. Since the structure function of a homogeneous MSS can be viewed as a MVL function, MDDs can also be used to represent the system structure function [10]. Furthermore, they can be modified in such way that they can be applied to store the structure function of non-homogeneous systems [11]. In what follows, we will focus on the modified version of MDD.

The modified MDD for representation of the structure function is a graph structure consisting of two types of nodes: sink (terminating) nodes and non-sink nodes. The sink nodes are labeled by numbers from 0 to m -1, and they agree with the possible states of the system. The non-sink nodes represent individual system components. If a non-sink node corresponds to component i, then it has m_i outgoing edges that are labeled by numbers from 0 to m_i -1, and they correspond to individual states of the i-th component. Finally, the MDD has one root node. Every path from the root to sink node labeled by number j agrees with one or more state vectors for which the function represented by the MDD takes value j.

For illustration, let us consider the simple service system from [7] that consists of 3 components and whose structure function is defined by Table II. This function can be expressed in the form of MDD (Fig. 1). Since the system has 4 possible states, the corresponding MDD has 4 sink nodes that represent states 0, 1, 2, and 3 of the considered system. The non-sink nodes agree with components 1, 2, and 3 of the system. The nodes representing components 1 and 2 have 2 outgoing edges labeled by numbers 0 and 1 because these components have only 2 possible states. Similarly, every node corresponding to component 3 has 4 outgoing edges because this component can be in one of 4 possible states (for transparency, we have join the edges routing to the same node). Next, every path from the root to a sink node corresponds to one or more state vectors. For example, the red dashed path goes through the 0-outgoing edges of nodes agreeing with components 1 and 2, what implies that it corresponds to state vectors in which these components are in state 0, i.e. to state vectors of the form of (0,0,.). Since this path leads into sink node labeled by 0, the system is in state 0 for all situations in which the 1-st and 2-nd component are in state 0. Similarly, the green solid line corresponds to state vector (1,1,1), and the sink node implies that the structure function takes value 2 for this state vector.

V. CALCULATION OF INTEGRATED DIRECT PARTIAL LOGIC DERIVATIVES BASED ON MULTI-VALUED DECISION DIAGRAMS

An algorithm for computation of a DPLD based on a MDD has been considered in [11]. The principal idea of the algorithm for calculation of DPLD $\partial\phi(j \to h)/\partial x_i(s \to r)$ is creation of two sub-diagrams from the MDD. The first sub-diagram will be composed of all paths that go from the root to sink node j and that leave every node corresponding to component i through edge labeled by s. Similarly, the second sub-diagram contains all paths from the root to sink node h that leave nodes agreeing with component i via edge labeled by number r. The intersection of these sub-diagrams agrees with the set of the nonzero values of the DPLD. This algorithm can be simply modified for finding the nonzero elements of IDPLDs. This modification lies in the fact that the first sub-diagram will be composed of all paths that leave the nodes corresponding to component i through edge labeled by s and that finish either in sink node labeled by j (for IDPLD $\partial\phi(j\downarrow)/\partial x_i(s \to r)$) or in a sink node corresponding to a state greater than j (for IDPLD $\partial\phi(\downarrow j)/\partial x_i(s \to r)$) or in a sink node corresponding to a state greater than or equal to j (for $\partial\phi(h_{\geq j} \to h_{<j})/\partial x_i(s \to r)$). In the similar sense, the second sub-diagram will consist of all paths that leave the nodes agreeing with the i-th component via edge labeled by r and that end either in a sink node corresponding to a system state less than j (for IDPLDs $\partial\phi(j\downarrow)/\partial x_i(s \to r)$ and $\partial\phi(h_{\geq j} \to h_{<j})/\partial x_i(s \to r)$) or in sink node labeled by j (for $\partial\phi(\downarrow j)/\partial x_i(s \to r)$). Furthermore, for calculation of IDPLD $\partial\phi(j\downarrow)/\partial x_i(s \to r)$, it can be useful to join all sink nodes labeled by numbers from set $\{0,1,\ldots,j\text{-}1\}$ into one node because this can result in simplification of the MDD used for finding the sub-diagrams. Similarly, in the case of IDPLD $\partial\phi(\downarrow j)/\partial x_i(s \to r)$, it is useful to join all sink nodes labeled by numbers $j+1, j+2, \ldots, m$ -1 into one node and, in the case of IDPLD $\partial\phi(h_{\geq j} \to h_{<j})/\partial x_i(s \to r)$, all sink nodes agreeing with system states $j, j+1, \ldots, m$ -1 into one node and all other sink nodes into another one.

TABLE II. STRUCTURE FUNCTION OF THE SYSTEM

Components states		x_3			
x_1	x_2	0	1	2	3
0	0	0	0	0	0
0	1	0	1	1	2
1	0	0	1	1	2
1	1	0	2	3	3

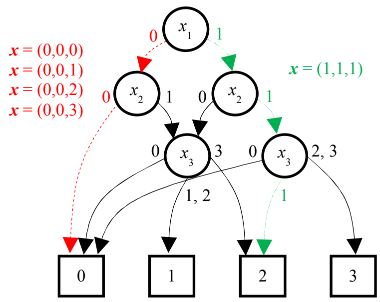

Fig. 1. MDD of the considered system.

For example, let us consider the MDD in Fig. 1, and let us assume that we want to find all nonzero elements of IDPLD $\partial\phi(3\downarrow)/\partial x_1(1 \to 0)$. This implies that we want to identify all situations in which change of component 1 from state 1 to 0 causes change of system state from value 3 to lower. Therefore, it is useful to join all sink nodes that are labeled by numbers lower than 3. This results in simplification of the MDD (Fig. 2). Next, based on definition (11), calculation of the considered IDPLD requires identifying all state vectors $(1_1, \boldsymbol{x})$ for which the system is in state 3 and all state vectors $(0_1, \boldsymbol{x})$ for which the system is in a state that is less than 3 (the left and middle sub-diagrams in Fig. 3). The sub-diagrams storing these state vectors can be obtained simply by traversing the MDD in Fig. 2. Finally, we have to find intersection of these two sub-diagrams (the rules for calculation of the intersection has been described in detail in [11]). The intersection is another sub-diagram (the right one in Fig. 3). This sub-diagram has only one sink node (leaf), and the root-leaf paths correspond to state vectors $(\cdot_1, \boldsymbol{x})$ for which change of component 1 from state 1 to 0 causes degradation of system state 3. The considered IDPLD contains 8 elements because there are 8 possible state vectors of the form of $(\cdot_1, \boldsymbol{x})$. Two of them are nonzero (the right part of Fig. 3), therefore, the truth density of this derivative is 0.25. Based on (14), this truth density agrees with the SI of state 1 of component 1 for degradation of system state 3. In the similar way, the MDD in Fig. 2 can be used to find all sub-diagrams that are needed for computation of IDPLDs of the form of $\partial\phi(3\downarrow)/\partial x_i(s \rightarrow s-1)$ that can be used to compute all other SI measures of the form of $\mathrm{SI}_{i,s}^{3\downarrow}$. The main advantage of the approach based on IDPLDs is that only two traversals of the MDD have to be performed to identify all situations in which state s of the i-th system component is critical for degradation of system state j (in case of DPLDs, $2(j-1)$ traversals have to be performed) and, therefore, the approach based on IDPLDs is more efficient than approaches based on DPLDs.

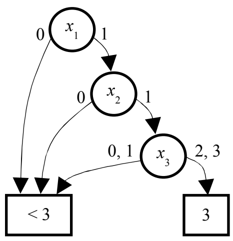

Fig. 2. MDD for calculation of IDPLDs $\partial\phi(3\downarrow)/\partial x_i(s \rightarrow r)$.

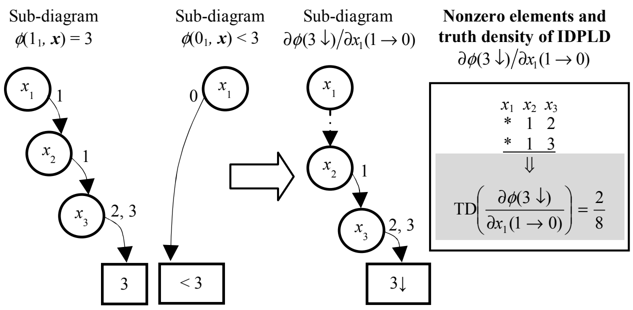

Fig. 3. Computation of IDPLD (11) based on MDD.

VI. Conclusion

One of the principal tasks of reliability analysis is investigation of importance of individual system components. Results of such analysis can be used to optimize system reliability/availability or to plan system maintenance. The problem is how to perform this analysis efficiently in the case of systems composed of many different components. We presented that classical DPLDs are not very appropriate for such analysis because they carry quite little information. Therefore, we proposed new types of DPLDs that were named as IDPLDs. These derivatives can be simply applied to any type of a coherent MSS to find situations in which a given state of a given component is critical for system degradation. However, the main issue is how to compute these derivatives when the system structure function is defined based on a MDD, which is very efficient for representation of complex systems. For this purpose, we proposed some modifications of the algorithm for calculation of DPLDs from [11].

References

[1] M. Rausand and A. Høyland, System Reliability Theory, 2nd ed. Haboken, NJ: John Wiley & Sons, Inc., 2004.

[2] A. Lisnianski and G. Levitin, Multi-state System Reliability. Assessment, Optimization and Applications. Singapore, SG: World Scientific, 2003.

[3] B. Natvig, Multistate Systems Reliability Theory with Applications. Chichester, UK: John Wiley & Sons, Ltd, 2011.

[4] Y. Watanabe, T. Oikawa, and K. Muramatsu, "Development of the DQFM method to consider the effect of correlation of component failures in seismic PSA of nuclear power plant," Reliability Engineering & System Safety, vol. 79, no. 3, pp. 265–279, Mar. 2003.

[5] B. Nystrom, L. Austrin, N. Ankarback, and E. Nilsson, "Fault tree analysis of an aircraft electric power supply system to electrical actuators," in International Conference on Probabilistic Methods Applied to Power Systems 2006 (PMAPS 2006), 2006, pp. 1–7.

[6] E. Zio, "Reliability engineering: Old problems and new challenges," Reliability Engineering & System Safety, vol. 94, no. 2, pp. 125–141, Feb. 2009.

[7] M. Kvassay, E. Zaitseva, and V. Levashenko, "Minimal cut sets and direct partial logic derivatives in reliability analysis," in Safety and Reliability: Methodology and Applications - Proceedings of the European Safety and Reliability Conference, ESREL 2014, 2015, pp. 241–248.

[8] E. N. Zaitseva and V. G. Levashenko, "Importance analysis by logical differential calculus," Automation and Remote Control, vol. 74, no. 2, pp. 171–182, Feb. 2013.

[9] E. Zaitseva, "Importance analysis of multi-state system by tools of differential logical calculus," in Reliability, Risk, and Safety. Theory and Application, vol. 3, C. Guedes Soares, R. Briš, and S. Martorell, Eds. London, UK: CRC Press, 2010, pp. 1579–1584.

[10] E. Zaitseva and V. Levashenko, "Multiple-valued logic mathematical approaches for multi-state system reliability analysis," Journal of Applied Logic, vol. 11, no. 3, pp. 350–362, Sep. 2013.

[11] E. Zaitseva, V. Levashenko, J. Kostolny, and M. Kvassay, "A multi-valued decision diagram for estimation of multi-state system," in Eurocon 2013, 2013, pp. 645–650.

[12] W. S. Griffith, "Multistate reliability models," Journal of Applied Probability, vol. 17, no. 3, pp. 735–744, Sep. 1980.

[13] S. Wu, "Joint importance of multistate systems," Computers & Industrial Engineering, vol. 49, no. 1, pp. 63–75, Aug. 2005.

[14] D. A. Butler, "A complete importance ranking for components of binary coherent systems, with extensions to multi-state systems," Naval Research Logistics Quarterly, vol. 26, no. 4, pp. 565–578, Dec. 1979.

[15] Z. W. Birnbaum, "On the importance of different components in a multicomponent system," in Multivariate Analysis, vol. 2, P. R. Krishnaiah, Ed. New York, NY: Academic Press, 1969, pp. 581–592.

[16] S. N. Yanushkevich, D. M. Miller, V. P. Shmerko, and R. S. Stankovic, Decision Diagram Techniques for Micro- and Nanoelectronic Design Handbook, vol. 2. Boca Raton, FL: CRC Press, 2005.

[17] D. M. Miller and R. Drechsler, "On the construction of multiple-valued decision diagrams," in Proceedings of the 32nd IEEE International Symposium on Multiple-Valued Logic (ISMVL 2002), 2002, pp. 245–253.