IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 2, 2017, accepted December 29, 2017, date of publication January 5, 2018, date of current version February 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2789931

Multi-State Reliability Assessment Method Based on the MDD-GO Model

YI REN, (Member, IEEE), CHENCHEN ZENG, DONGMING FAN, LINLIN LIU, AND QIANG FENG, (Member, IEEE)
School of Reliability and Systems Engineering, Beihang University, Beijing 100191, China
Corresponding author: Qiang Feng (fengqiang@buaa.edu.cn)

ABSTRACT The GO methodology is an important technique for reliability assessment and evaluation of complex systems, especially for systematic timing and dynamic characteristics. However, the treatment of the shared signal and multi-state issues has increased the calculation burden in GO mode. Multi-valued decision diagrams (MDDs) have the merit of effectively performing qualitative and quantitative analysis via Boolean function forms directly; hence, in this paper, we propose a novel efficient algorithm for the multi-state GO model based on MDD, which combines the advantages of the two techniques. Moreover, the algorithm avoids the complex separate process of handling shared signals in the probability formulas method and the problem of complicated calculation in the state compounding method. In addition, the path and cut sets of the system are obtained through qualitative calculation based on the new algorithm. Finally, the correctness, simplicity, and accuracy of the new algorithm for analyzing systems that contain shared signals are verified by a case study.

INDEX TERMS Reliability assessment, multi-valued decision diagrams, GO methodology, multi-state model.

I. INTRODUCTION

The GO methodology [1], which is a success-oriented approach for probability analysis of a system, has a strong description capability for modeling the reliability and safety in multi-state, time-related systems, especially for a system with matter flow, such as air, liquid and current [2]. In reliability assessment, a variety of multi-state components broadly exist in systems [3], [4], and the current GO methodology and its extension GO-FLOW have been applied in several fields, which incorporates the flowing information into the signal flow graph, signals represent physical quantities or information, e.g., working time, nuclear energy safety issues [5]. The basic idea behind the GO methodology (developed on the basis of decision trees) is that a design principle diagram or a work-flow-chart can be transformed into a GO chart easily. In addition, the operators and signal flow of the GO methodology can represent the multiple states of a system [6]; thus, multi-state probability analysis of a system can be obtained directly. The GO methodology theory and algorithm will be improved and developed further in the future during the process of its popularization [7]. To date, there are two primary quantitative analysis methods of the GO methodology, i.e., state combination algorithm and probability formula algorithm. The limitation of the second method mainly involves overly complicated calculation, which is reflected in two aspects: a correction formula results in additional calculation when shared signals exist, and the calculation of the cumulative state probability is too onerous when faced with a multi-state system (MSS).

To solve these existing issues, several improved methods have been proposed after investigation of the weakness of the existing algorithms. For example, the probability matrix algorithm (PMA) [8], [9], which is based on the probability formula algorithms, settles the correction problem regarding the shared signal by listing the matrix equation without the restriction of the state dimension. In addition, several scholars focus on technique combination instead of arithmetic improvement. For example, they proposed that the GO methodology can be combined with other models, such as Bayesian Networks (BNs) methods or dynamic BNs [10], [11], to simplify the computing problem; such an algorithm based on BNs was proposed in [12] and [13]. Specifically, the mature algorithm software, which can calculate the posterior probability distribution of nodes, and the ability of BNs to describe event polymorphism have enabled BNs to be widely applied in the reliability field recently [14].

The above method of converting GO to BNs shows a strong effect on the quantitative analysis in reliability assessment; however, the academic discussion on qualitative analysis, e.g., determining how to obtain minimal cut and set paths efficiently, has seldom been mentioned. Therefore, in this paper, we propose the combination of the GO methodology with an appropriate multi-state model to simultaneously increase the efficiency in the quantitative and qualitative analysis.

Regarding the quantitative analysis methods of probability in an MSS, there are other multi-state models besides the Bayesian Network, e.g., multi-state fault tree (MFT) [15], multi-valued minimal paths [16] and multi-valued decision diagrams (MDD). Essentially, MDD is an extension of BDD [17], [18], the concept was first proposed by Miller [19]. The binary decision diagram (BDD), based on the directed acyclic graph (DAG) of Shannon's decomposition, is a graph-based expression of Boolean functions that can strongly reduce the computational burden and improve both the calculation efficiency and calculation accuracy but with the limitation that it cannot handle the multi-state problems of the GO Methodology [20]. MDD has been widely applied in many fields and system like MSS at present and is an effective method to store and compute a multi-state model as well as obtain both cut and set paths. Especially, MSS analysis of reliability or safety like minimal cut sets (or minimal cut vectors (MCVs) which is important in reliability engineering can be accomplished by methods of MDD referring to the multiple-valued logic [21]–[23], or topological analysis based on direct partial logic derivatives [24]. Several studies on MDD can be found in the literature. For example, Mo proposed a novel reliability analysis algorithm in MSS [25] by building the MDD of multi-state k-out-of-n systems to reduce the computational complexity. In addition, Zhai utilized MDD to perform reliability and complexity analysis of the k-out-of-n system containing n-k cold storage units [26]. Besides, MDD method proposed by Elena Zaitseva is an efficient approach for the examination analysis of the MSS structure function of high dimension is proposed [27], [28]. To combine the advantages of the GO methodology and MDD together, in this article, we propose an MDD-based method, called the "MDD-GO model", to solve the performance problem. Converting the GO chart into MDD is implemented because the MDD approach is an effective method to compute a multi-state model and can be used to perform direct qualitative and quantitative analysis, thereby fundamentally avoiding the following issues: (i) the complex separate process of handling shared signals in the probability formulas method and (ii) the problem of state explosion in the state compounding method in the GO methodology.

The rest of this paper is organized as follows. Section II primarily introduces background knowledge of the GO method and MDD. Section III provides a description of our detailed work in this paper, involving the new multi-state GO algorithm based on MDD. Section IV describes an illustrative example. In section V, our conclusions and results are presented, as well as orientation for our future research. TABLE I below shows the expansion of nomenclature mentioned in the paper.

II. PRELIMINARY CONCEPTS

A. GO METHODOLOGY AND RELATED ALGORITHMS

GO-FLOW and the GO Methodology [29], [30] are based on the most basic elements of signals and operators that build a complete GO chart from a systematic principle diagram, flow chart and engineering drawings [1]. There are 17 types of operators divided into two broad categories: the logical operators and the functional operators. The first type of operator represents an arithmetic logic, nevertheless, such an operator has a characteristic that it has no state itself relative to input or output signal. The second type of operator has the function state itself and arithmetic logic simultaneously compared with the logical operators. To be more precise, operator classes 2, 9, 10, 11, 13, 14, and 15 are logical operators; and operator classes 1, 3, 4, 5, 6, 7, 8, 12, 16, and 17 are functional operators. Among these operators, the most common classes in engineering are 1, 2, 3, 5, 6, 7, 9, 10, 11, and 15; these typical operators are studied in this paper to show how the MDD models are generated using the proposed algorithm [31].

The traditional GO Methodology has two main types of quantitative algorithms: probability formulas method [1] and state compounding method. The following context provides a brief introduction to the prior method. The concept of the probability formulas method was proposed by Zu-pei Shen in 2000; the method can precisely handle the computational problem when the system contains shared signals, and the method was improved in 2000 [1]. The state probability of the output signals is produced by substituting the probabilistic data of the operators and the input signals into the probability formulas; this approach, skillfully avoids the process of listing all the state combinations. Therefore, the method to a great extent reduces the calculation amount of the GO methodology. Despite these advantages for the related algorithms, the proposed MDD-based method has more efficiency in computation than the existing recursive method. The precision processing method of shared signal involves correction of the state probability calculation formula. Accumulation probability of the state of the output signal A_R is represented by a polynomial function: $A_R = N(A_{S_1}, A_{S_2}, \ldots, A_{S_M})$, and $A_{S_1}, A_{S_2}, \ldots, A_{S_M}$ represent the accumulation probability of the respective input signals. If the input signal contains one shared signal S_a, then A_{S_j} and A_R can be respectively shown as:

$$A_{s_j} = a_{0j} + a_{1j}A_{s_a}$$
$$A_R = (1 - A_{s_a})a_0 + A_{s_a}\sum_{j=0}^{M} a_j \tag{1}$$

Here a_{0j} and a_{1j} are no associated with shared signal S_a, $a_{1j} \neq 0$ means that S_j has relationship with S_a and $a_{0j} = 0$ means that S_j completely contains S_a. As well, the formula to solve the accumulation probability of the state of output signal A_R has been corrected.

TABLE 1. Explanation of nomenclature mentioned in the paper.

Nomenclature		Nomenclature	
N	Number of state of components	G, H	Decision diagrams
V	Node space in MDD	S, C	Root node of MDD mapped from function operator's input signal and operator
x_i	The component stays at state i but not j	S'	Root node of MDD mapped from logic operator's input signal
$F_{x_A=j}$	The multi-valued logic functions of A	R	Output signal in operator
F	Case format of a logic expression	BDD	Binary decision diagram
$\varphi(x)$	System atate structure function, $0 \le \varphi(x) \le n$	MDD	Multiple-valued decision diagram
$value(V_i)$	Node value in MDD_i	MDD_i	MDD of system in cumulative state

If input signal contains L shared signals, S_l, $(l = 1, \ldots, L)$, and the corresponding accumulation probability of state is A_{S_l}, then the accumulation probability of the state of system A_R is shown as:

$$A_R = \sum_{K_1=0}^{1} \sum_{K_2=0}^{1} \cdots \sum_{K_L=0}^{1} A_{RK_1K_2 \ldots K_L} \times \prod_{l=1}^{L} [(1 - A_{s_l})(1 - K_l) + A_{s_l} K_l] \quad (2)$$

Where $\mathrm{A_{RK_1K_2 \ldots K_L}}$ represent the system's value of accumulation probability when L shared signals are in a certain combined state.

B. MULTI-VALUED DECISION DIAGRAM (MDD)

BDD was first proposed by Akers in 1978 [32], [33]; its computation requirement depends linearly on the graphic scale of BDD, and the linearity effectively solves the problem that the computation requirement in the traditional method of fault tree analysis (FTA) grows exponentially with its graphic scale. In the aspect of qualitative analysis, we can obtain the minimum cuts set according to the BDD chart, and the top event's accurate failure probability can be quantitatively calculated. Moreover, the events of FTA are non-coherent; even for BDD, the assessment of such large formulas is challenging [34].

MDD, an extension of BDD, is an existing and effective graph-based data structure for symbolic representation and manipulation of discrete logical functions with multiple values. In form, based on Shannon's decomposition theorem, MDD can serve as logical functions as rooted, directed acyclic graphs that are both authoritative and compact. MDDs have found widespread use in applications of reliability, such as formal circuit verification, logic synthesis, test generation, and re-synthesis for network optimization in a system.

MDD is symbolized by (V, M), and the nodes are divided into two types: sink node and non-sink node. More specifically, an MDD consists of decision nodes and two sink nodes, which are labeled 0 and 1($value\,(v) \in \{0, 1\}$); the labels 0 and 1 indicate that the system is not and is in a specific state, respectively. Here, we consider the particular state as

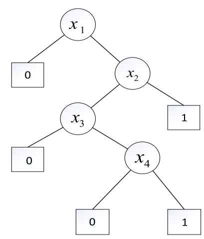


FIGURE 1. MDD schematic.

the successful state of system and the component A of a generalized multi-state system (MSS) [35], [36] has a space of state with n states, $j \in \{1, 2, \ldots, n\}$. Let x be the state indicator variable of this component A, where $x_A = i$ means denotes the part that remains at state i but not at state $j \neq i$. Each non-sink decision node has multiple directed edges of the outgoing signal when the ith-edge $(0 \le i \le n)$ connects to the child node, which is the definition of the multi-valued logic functions $F_{x=i}$. Thus, each non-sink node in the MDD encodes a case construct as shown below. The determined index values M are always a concern and are generally solved by experience.

For example, as shown in Fig. 1, MDD has $V = \{x_1, x_2, x_3, x_4, 1, 1, 0, 0, 0\}$, $index\ M = \{1, 2, 3, 4\}$, $index\,(x_1) = 1$, $index\,(x_2) = 2$, $index\,(x_3) = 3$, $index\,(x_4) = 4$.

To easily describe the manipulation of the MDD, the case format of a logic expression F is defined for an MSS expanded with regard to a component A with n-state (which is denoted by $1, 2, \ldots, n$):

$$\begin{aligned} F &= A_1 \cdot F_{x_A=1} + A_2 \cdot F_{x_A=2} +, \ldots, A_n \cdot F_{x_A=n} \\ &= case(A, F_{x_A=1}, F_{x_A=2}, \ldots, F_{x_A=n}) \\ &= case(A, F_1, F_2, \ldots, F_n \end{aligned} \quad (3)$$

Fig.2. represents the general format of the MDD of this expression and the particular format when the MDD contains a basic event, A_i (A being in state i).

An MDD manipulation can represent a logic operation between component G and H [37], [38]. Consider that two component-level MDD models with top MDD nodes G and

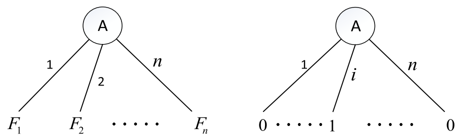


FIGURE 2. MDD model.

H can be represented by the following case format, and X and Y are the names of components corresponding to MDD.

$$G \circ H = case(x, G_1, G_2, \ldots, G_n) \circ case(y, H_1, H_2, \ldots, H_n)$$
$$= \begin{cases} case(x, G_1 \circ H_1, G_2 \circ H_2, \ldots, G_n \circ H_n) \\ \quad index(x) = index(y) \\ case(x, G_1 \circ H, G_2 \circ H, \ldots, G_n \circ H) \\ \quad index(x) < index(y) \\ case(y, G \circ H_1, G \circ H_2, \ldots, G \circ H_n) \\ \quad index(x) = index(y) \end{cases} \tag{4}$$

In the equation (4), $\circ$ represents any logic operations, with & and *OR*, labeled "$\cdot$" and "+", respectively. Obviously, $0 < OR > H = H$, $1 < OR > H = 1$, and $0 < \& > H = 0$, $1 < \& > H = H$. In this paper, we only consider the & and *OR* logic operations because the "*k*-out-of-n" operator can be translated into these two categories of logic operations.

C. ACCUMULATION PROBABILITY OF STATE

Introducing the concept of accumulation probability [1] can allow the calculation of the output signal's state probability based on input signal's state probability and operator. Thus, one can directly determine the probability in each state in the system.

In general, the signal flow of the multi-state GO model has $N+1$ state values, $0, 1, \ldots, N$, where 0 represents the advanced state, and $1, \ldots, N-1$ represent multiple states; N also represents the failure state. In this paper, we do not consider the special condition when state value is 0.

The probability of the input signal in state i, $P_s(i)$, is computed quantitatively in MDD; thus, the probability of output signal in state i, $P_R(i)$ can thus be derived. $\sum_{i=1}^{n} P_R(i) = 1$, $\sum_{i=1}^{n} P_s(i) = 1$. Because of the independence between states, and because the sum of the state probabilities is 1, the cumulative probability of input and output signal in state i, marked $A_S(i)$ and $A_R(i)$ [39], are as follows:

$$A_S(i) = \sum_{j=1}^{i} P_S(j) \quad i = 1, \ldots, N-1 \; A_S(N) = 1$$
$$A_R(i) = \sum_{j=1}^{i} P_R(j) \quad i = 1, \ldots, N-1 \; A_R(N) = 1 \tag{5}$$

The equation (5) shows that $A(i)$ is the sum of signal flow's status probabilities, whose status value is 1,2…i, and $P_R(i)$, the state probability of output signal in particular state i, is determined by the equation (6):

$$P_R(1) = A_R(1)$$
$$P_R(i) = A_R(i) - A_R(i-1) \quad i = 2, \ldots, N \tag{6}$$

$A_R(i)$, the cumulative probability of output signal is calculated, then $P_R(i)$ can be deduced by $A_R(i)$.

III. MDD-BASED APPROACH

The new algorithms for the multi-state GO model based on MDD can be applied and have the following process: (i) defining mapping rules from frequently used GO operators to MDD; (ii) giving the conversion algorithm process of mapping complete GO model to MDD; and (iii) making the quantitative and qualitative analysis regarding the minimum cuts, path sets and cumulative probability in *i*th-state based on the MDD, and constructing the quantitative calculation process and cut set generation algorithm. The detailed contents of these steps are described in the following subsections.

A. OPERATOR GENERAL MAPPING RULES

The specific operation for mapping some frequently used GO operators (including input signal, operator itself and output signal) to MDD is as follows:

Step 1: if the operator is the function operator, then map the input signal and the operator itself to the root node of MDD separately, described by S and C, respectively; or else, we only map the input signal of logical operator and describe it as S';

Step 2: S or S' has n leaf nodes, $value\,(x_i) = 1$ and $value\,(x_j) = 0\ (j \neq i)$; C has leaf nodes of values 0 or 1;

Step 3: map the output signal to MDD, described by R; if the operator is the function operator, then $R = S_1 \circ S_2 \circ \ldots S_i \circ C$ or else $R = S'_1 \circ S'_2 \circ \ldots S'_i$.

Apparently, the output signal state is determined by the operators logical operations between the input signal and the operators own state (except for the logical operators); hence, the relationships among the operator input signal, the operator state itself and output signal's state can be intuitively described by the probability in each state combination table of the operator logic.

Operators class: 2(*OR* logic), 10(& logic), 11(*k*-out-of-*n*) [40], [41] are contained in the logical operators; Operators class: 1 (two-state units), 5 (signal generator), 6 (conduction element with signal) are contained in the functional operators. In this paper, we chose these typical operators listed above to illuminate the derived rule.

The next two sections will combine the abovementioned mapping rules to discuss the conversion process for these typical operators in detail.

1) FUNCTION OPERATOR MAPPING TRANSFORMATION

a: "TYPE 1 TWO-STATE UNITS"

The type 1 operator is most commonly used in engineering modeling; as shown in Fig. 3, it has one input signal and output signal with a two-state operator. Both the input signal

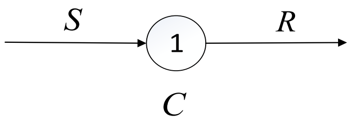


FIGURE 3. Operator of type 1.

TABLE 2. Operationl rule of type 1.

V_S	V_C	V_R
$1,\ldots,N-1$	1	$1,\ldots,N-1$
N	1	N
$1,\ldots,N-1$	2	N

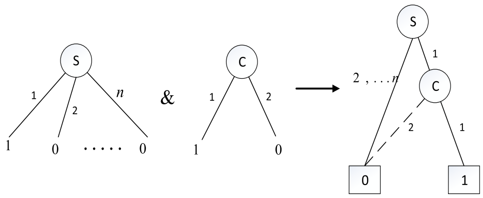


FIGURE 4. MDD_1 of the type 1 operator.

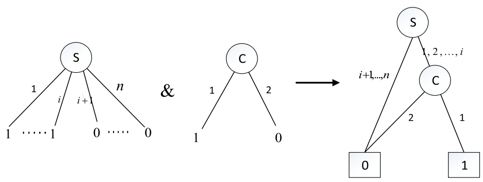


FIGURE 5. MDD_i of the type 1 operator.

and the operator should be mapped into nodes in MDD, and the logical between them is the & operation. V_S is the status value of the input signal, V_C is the status value of the operation, and V_R is the status value of the output signal.

The operator's operation logic is shown in TABLE 2.

Following the mapping transformation rules mentioned above, the accumulative state 1 and i of the type 1 operator, MDD_1 and MDD_i are shown in Fig. 4 and Fig. 5 respectively.

The following presents the calculation process.

$$\begin{aligned} MDD_1 &= case(s, case_1(1,0), 0, \ldots, 0) \\ MDD_i &= case(s, case_1(1,0), case_2(1,0) \\ &\quad , \ldots, case_i(1,0), 0, \ldots, 0) \end{aligned} \tag{7}$$

b: "TYPE 5 SIGNAL GENERATOR"

Type 5 operator is shown in Fig. 6; as the system input has no input itself, it is a signal given by another system or system-dependent external event. V_R is the output signal statue value, $P_R(i)$ is the probability of the ith-state, and $\sum P_R(i) = 1$.

The same goes for type 5; MDD_i of the type 5 operator is shown in Fig. 7.

$$MDD_i = case(s, 1, \ldots, 1, 0, \ldots, 0) \tag{8}$$

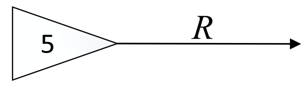


FIGURE 6. Operator of type 5.

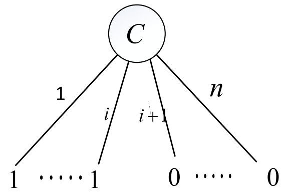


FIGURE 7. MDD_i of type 5 operator.

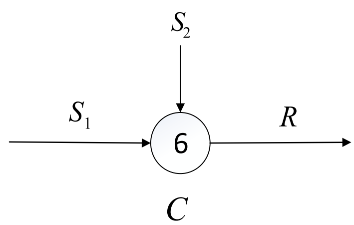


FIGURE 8. Operator of type 6.

TABLE 3. Operationl rule of type 6.

V_{S1}	V_{S1}	V_C	V_R
$I_1(1,\ldots,N)$	$I_2(1,\ldots,N)$	1	$\max(I_1, I_2)$
$I_1(1,\ldots,N)$	$I_2(1,\ldots,N)$	2	N

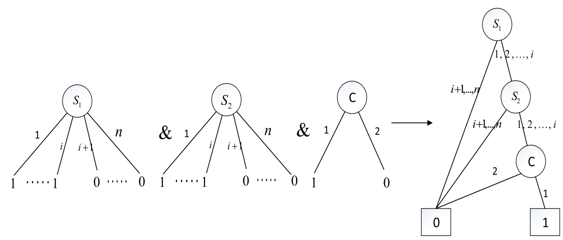


FIGURE 9. MDD_i of the type 6 operator.

c: "TYPE 6 CONDUCTION ELEMENT WITH SIGNAL"

Type 6 operator, as shown in Fig. 8, is used to simulate a component that has two input signals S, $S \in \{S_1, S_2\}$. The signal status value is $V_{S_i} \in \{1, 2, \ldots, N\}$, the type 6 operator itself has two states, $V_C \in \{1, 2\}$.

The operator's operation logic is shown in TABLE 3.

Similarly, MDD_i is shown in Fig. 9.

$$\begin{aligned} MDD_i = case(&s_1, case_1(s_2, case_1(c, 1, 0), \ldots, \\ &case_i(c, 1, 0), 0, \ldots, 0), \ldots, case_i(s_2, case_1 \\ &(c, 1, 0), \ldots, case_i(c, 1, 0), 0, \ldots, 0), 0, \ldots, 0) \end{aligned} \tag{9}$$

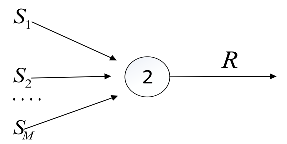


FIGURE 10. Operator of type 2.

TABLE 4. Generation alforithm for $\mathbf{MDD_{G_i}}$.

Number	Algorithm to generate the MDD model MDD_{G_i}
0	MDD_{G_i} Gen(index, i, M)=*/index= BDD_{G_i} Gen(G_i, n)*/
1	For each case x, $0 \le x \le index - 1$
2	$MDD[x]$.name= $BDD[x]$.name
3	For each case y, $0 \le y \le i - 1$
4	$MDD[x]$.Bra(y)= $BDD[x]$.E
5	For each case y, $i \le y \le M$
6	$MDD[x]$.Bra(y)= $BDD[x]$.T

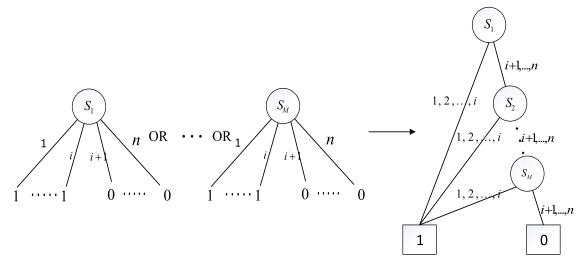


FIGURE 11. $\mathbf{MDD_i}$ of the type 2 operator.

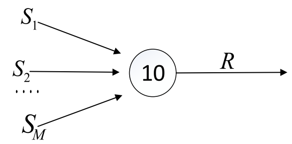


FIGURE 12. Operator of type 10.

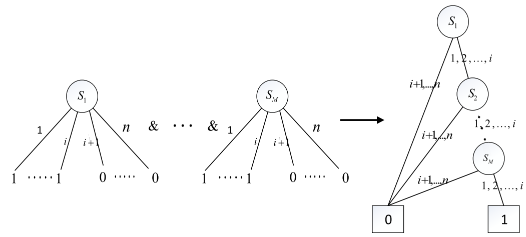


FIGURE 13. MDD_i of the type 10 operator.

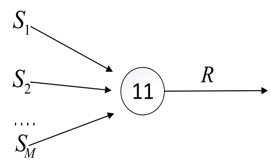


FIGURE 14. Operator of type 11.

2) LOGICAL OPERATOR'S MAPPING TRANSFORMATION

a: "TYPE 2 OR LOGIC"

The type 2 operator corresponds to *OR* logic with multiple input signal and one output signal both having multiple states. The output signal is $V_R = min\{V_{S_1}, V_{S_2} \dots V_{S_M}\}$.

We illustrate this operation with an example of $M = 3$ operator of type 2 is shown in Fig. 10 above; the MDD_i is shown in Fig. 11.

$$\begin{aligned} MDD_i = case(&s_1, 1, \dots, 1, case_{i+1}(s_2, 1, \dots, 1, \\ &case_{i+1}(s_3, 1, \dots, 1, 0, \dots, 0), \dots, \\ &case_n(s_3, 1, \dots, 1, 0, \dots, 0)), \dots, \\ &case_n(s_2, 1, \dots, 1, case_{i+1} \\ &(s_3, 1, \dots, 1, 0, \dots, 0), \dots, \\ &case_n(s_3, 1, \dots, 1, 0, \dots, 0))) \end{aligned} \quad (10)$$

b: "TYPE 10 & LOGIC"

Type 10 operator is similar to type 2, and the output signal is $V_R = max\{V_{S_1}, V_{S_2} \dots V_{S_M}\}$.

We illustrate this operation with an example of $M = 3$ and operator of type 10 is shown in Fig. 12 above; the MDD_i is shown in Fig. 13.

$$\begin{aligned} &MDD_i \\ &= case(s_1, case_1(s_2, case_1(s_3, 1, \dots, 1, 0, \dots, 0) \\ &\quad , \dots, case_i(s_3, 1, \dots, 1, 0, \dots, 0), 0, \dots, 0), \dots, \\ &\quad case_i(s_2, case_1(s_3, 1, \dots, 1, 0, \dots, 0), \dots, \\ &\quad case_i(s_3, 1, \dots, 1, 0, \dots, 0), 0, \dots, 0), 0, \dots, 0) \end{aligned} \quad (11)$$

c: "TYPE 11 k-OUT-OF-n"

Types 2 and 10 are special cases of the type 11 operator, which first sorts an array of input signals from small to large and subsequently selects the kth-signal as the output signal, $V_R = V_K$.

Based on the manipulation rule given above and operator of type 11 is shown in Fig. 14 above, TABLE 4 gives the algorithm to generate the MDD model $MDD_{G_i}(G_i \in \{G_1, G_2 \dots G_M\})$, derived by BDD_{G_1}, $BDD_{G_2} \dots BDD_{G_M}$ via the process of incorporating the multi-state message in the system. During the process of implementation, a table MDD stores all the nodes of the MDD. The grouping of the nodes as a tuple is named as a collection: $Bra \in \{Bra(0), Bra(1), Bra(2), \dots, Bra(M)\}$, such that $Bra(i)$ connected to the ith-edge of this node is an import of the MDD node. When $(x == n - G_i)$, and $E = ZERO$, else, $E = index + 1$; when $(y == G_i - 1)$, and $T = ONE$, else, $T = index + n - G_i + 1$. When the MDD nodes include sink nodes that are represented by values 0 and 1, we can utilize *ZERO* and *ONE* to represent them.

MDD_{G_1}, MDD_{G_2} and MDD_{G_3} are shown in Fig. 15; the $k - out - of - 5$ structure is shown for states 1, 2, and 3, where $G_1 = G_2 = G_3 = 3$.

B. MDD CONVERSION MAPPING RULES

We present the common operators' mapping rules in section III, and the algorithm flowchart converting complete GO chart into MDD is discussed further in

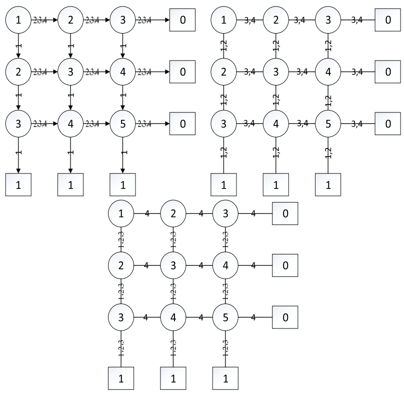


FIGURE 15. MDD_{G_1}, MDD_{G_2} and MDD_{G_3} of the type 11 operator.

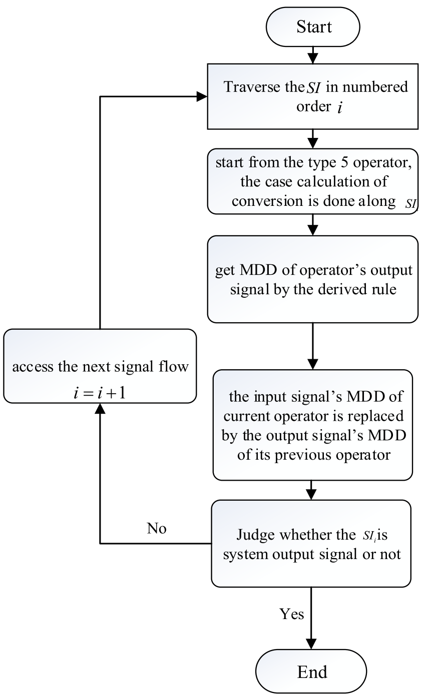


FIGURE 16. Flowchart of the mapping rule.

this chapter. The flowchart is shown in Fig. 16 and is described below:

Step 1: we traverse the signal flow collection $SI = \{SI_1, SI_2 \ldots SI_N\}$ in numbered order i, starting from the type 5 operator, and then the case calculation of conversion is performed along SI_i;

Step 2: obtain the MDD of the operator's output signal in each accumulative state using the derived rule listed in section III;

Step 3: during the case calculation, the input signal's MDD of the current operator is replaced by the output signal's MDD of its previous operator;

Step 4: judge whether the signal flow SI_i is a system output signal or not ($i = N$ or $i \neq N$),); if not, then access the next signal flow SI_{i+1}, otherwise, end the process and finish the mapping from the complete GO chart to the MDD.

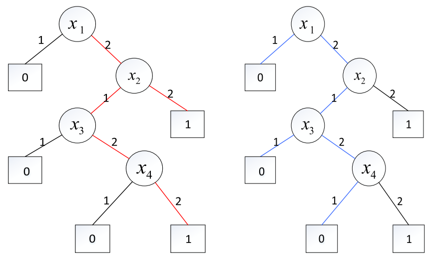


FIGURE 17. Path and cut sets in the MDD.

C. QUALITATIVE AND QUANTITATIVE ANALYSIS

1) QUALITATIVE ANALYSIS

The minimum path and cut sets of the system can be obtained through qualitative calculation based on the new algorithm. In this instance, we follow the steps outlined below to obtain the minimal path and cut sets, ps_i and cs_i, in each accumulative state i of the GO chart based on MDD_i.

Step 1: search all the paths in the MDD from root to leaf nodes when $value = 1$ in MDD_i $(i = 1, 2 \ldots N)$;

Step 2: obtain the path sets p_i, and then merge p_i to obtain the minimal path sets ps_i;

Step 3: analogously obtain the minimal cut sets cs_i.

For example, in Fig. 17, we can obtain the following path sets A and cut sets B:

$$\begin{aligned} &A\{x_1(2) - x_2(2), x_1(2) - x_2(1) - x_3(2) - x_4(2)\} \\ &B\{x_1(1), x_1(2) - x_2(1) - x_3(1), x_1(2) - x_2(1) \\ &\quad -x_3(2) - x_4(1)\} \end{aligned} \tag{12}$$

2) QUANTITATIVE ANALYSIS

The recursion method based on Shannon decomposition is a common method to quantitatively analyze the system MDD.

A recursive formula can calculate each state's cumulative probability of the system:

$$P(F) = p_{x_A=1}P(F_{x_A=1}) + p_{x_A=2}P(F_{x_A=2}), \ldots, p_{x_A=n}P(F_{x_A=n}) \tag{13}$$

$p_{x_A=i}$: the probability that part A is in the i-state; $P(F)$ is the occurrence possibility of MDD_F. The recursive formula's termination conditions is as follows: when $F_x = 0$; then $P(F_x) = z$; when $F_x = 1$, then $P(F_x) = 1$.

If the input signal of the GO model is defined as a shared signal, and the correcting algorithm as mentioned in section II should be introduced to amend the probability calculation formula of the operators. The inevitable correction occurs frequently and increases the work required remarkably because the system usually contains more than one shared signal. However, the new algorithm based on the MDD can avoid the problem; the quantitative analysis is based on the decision graph MDD, which is the supporter of probability expression. During the generative process of the MDD from the GO chart, the shared signal, which serves as the input signal of several operators, is considered independently.

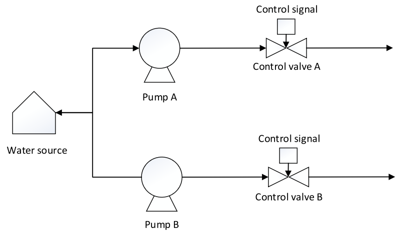


FIGURE 18. Two-branch water supply system.

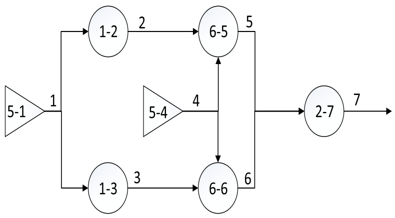


FIGURE 19. GO chart of the water supply system.

TABLE 5. Operator data of a two-branch water supply system.

Operator number	Operator type	Component	State probability		
			SV1	SV2	SV3
1	5	Water source	0.95	0.04	0.01
2	1	Pump A	0.99	0.01	0
3	1	Pump A	0.95	0.05	0
4	5	Control signal	0.9	0.05	0.05
5	6	Control signal A	0.99	0.01	0
6	6	Control signal B	0.98	0.02	0

sv1, sv2, sv3: respectively means status value of operators

IV. CASE STUDY

As an illustration, consider the system of a two-branch water supply. As shown in Fig. 18, the system consists of one water source and two pipe branches. The system will operate normally when at least one branch of pipes offers a normal water supply, and the control signal in the system is a shared signal because these two control valves are both driven by one control signal.

The GO chart of the water supply system is shown in Fig. 19; both the system and the operators defined in TABLE 5 have three states. Note that operators type 1 and 6 have states $\{1, 2\}$ and $\{0, 1, 2\}$, respectively, and the special state of 0 will not be considered; thus, the probability is 0 when type 1 and 6 operators are in the state of 3. Using the prior probabilistic formula algorithm, the state probability of operator number 7, which is also the state probability of system, is obtained, as shown in TABLE 6. During the calculation of the probabilistic formula algorithm, two shared signals, operator number 1 and 4, are used. The whole calculation process is extensive, except the precision processing method of correcting the shared signal, as described above; much tedious work is required to calculate each operator's state probability in turn based on the GO chart.

Next, we perform a quantitative analysis to determine the system's probability in the j-state, $j \in \{1, 2, 3\}$; here,

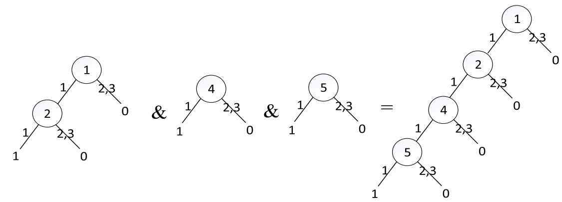


FIGURE 20. Export process of MDD_1 of output signal 5.

TABLE 6. Computation of the state probability of operator 7.

Input signal 5			Input signal 6			Input signal 7		
sv	sp	cp	sv	sp	cp	sv	sp	cp
1	0.8380	0.8380	1	0.7960	0.7960	1	0.8538	0.8538
2	0.0838	0.9218	2	0.080	0.8760	2	0.0867	0.9405
3	0.0782	1.0	3	0.124	1.0	3	0.0595	1.0

sv: status value of the operator; *sp*: state probability; *cp*: cumulative probability

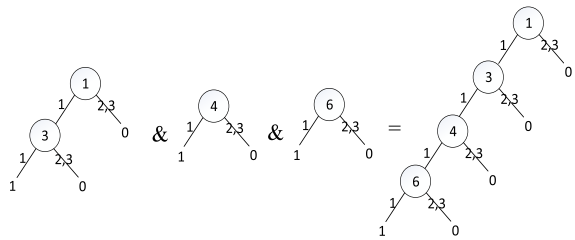


FIGURE 21. Export process of MDD_1 of output signal 6.

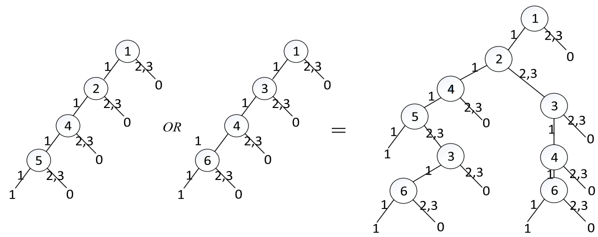


FIGURE 22. Export process of MDD_1 of output signal 7.

we present as an example the export process of MDD_1 of the output signal 7 (system output signal) using the mapping rule; the steps are shown in Fig. 20 to Fig. 22, and system MDD_1 and MDD_2 are shown in Fig. 23.

Sink nodes in the MDD labeled 0 and 1 indicate the system is not in a particular state and is in a particular state, respectively, and the probabilities that the water supply is in state j or below, $p\phi(x) \le j$, are given as follows:

$$p\{\phi(x) \le 2\} = p\{value(v_2) = 1\} = 0.9405$$
$$p\{\phi(x) \le 1\} = p\{value(v_1) = 1\} = 0.8538 \quad (14)$$

The state distribution of the system, $p\phi\,(x) = j$, can thus be deduced as follows:

$$p\{\phi(x) = 1\} = p\{\phi(x) \le 1\} = 0.8538$$
$$p\{\phi(x) = 2\} = p\{\phi(x) \le 2\} - p\{\phi(x) \le 1\} = 0.0867$$
$$p\{\phi(x) = 3\} = 1 - p\{\phi(x) \le 2\} = 0.0595 \quad (15)$$

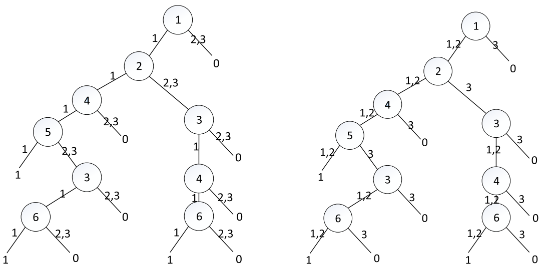


FIGURE 23. MDD of the water supply system.

TABLE 7. Path sets of the system in MDD_1 & MDD_2.

number	p_1	number	p_2
1	1(1)-2(1)-3(1)-4(1)-5(1)	1	1(1,2)-2(1,2)-3(1,2)-4(1,2)-5(1,2)
2	1(1)-2(1)-3(1)-4(1)-5(2,3)-6(1)	2	1(1,2)-2(1,2)-3(1,2)-4(1,2)-5(3)-6(1,2)
3	1(1)-2(1)-3(2,3)-4(1)-5(1)	3	1(1,2)-2(1,2)-3(3)-4(1,2)-5(1,2)
4	1(1)-2(2,3)-3(1)-4(1)-6(1)	4	1(1,2)-2(3)-3(1,2)-4(1,2)-6(1,2)

TABLE 8. Minimal path sets of the system in MDD_1 & MDD_2.

number	p_{S1}	number	p_{S2}
1	1(1)-2(1)-4(1)-5(1)	1	1(1,2)-2(1,2)-4(1,2)-5(1,2)
2	1(1)-3(1)-4(1)-6(1)	2	1(1,2)-3(1,2)-4(1,2)-6(1,2)

The above results agree well with the calculation shown in TABLE 6 using the corrected probability formula, this agreement highlights the correctness of utilizing the algorithm for the multi-state GO methodology based on MDD.

According to the qualitative analysis listed in section III, we can obtain the minimal path and cut sets, ps_i and cs_i. TABLE 7 and TABLE 9 list the path sets and cut sets, respectively, from root node, operator number 1, to leaf node when $value = 1$ or $value = 0$ based on the MDD_1 and MDD_2. As shown in TABLE 8 and TABLE 10, we found that the system in accumulative states 1 and 2 have two minimal path sets and six minimal cut sets.

The method based on MDD-GO model is novel and efficient for the multi-state GO model based on MDD, which combines the advantages of the two techniques. The method has made improvement in the matter of quantitative and qualitative analysis when analyze MSS, in this example, it avoids the correcting process about shared signal operators 1 and 4 and the complicated calculating of operators' probability in each state (the state probability of operator number $1, 2, 3, \ldots, 6$ is computed one by one in traditional corrected probability formula). Besides, path and cut sets can be obtained directly by analyzing MDD, the acquisition process is concise and explicit compared with the previous state

TABLE 9. Cut sets of the system in MDD_1 & MDD_2.

number	p_1	number	p_2
1	1(2,3)	1	1(3)
2	1(1)-2(1)-3(1)-4(2,3)	2	1(1,2)-2(1,2)-3(1,2)-4(3)
3	1(1)-2(1)-3(1)-4(1)-5(2,3)-6(2,3)	3	1(1,2)-2(1,2)-3(1,2)-4(1,2)-5(3)-6(3)
4	1(1)-2(2,3)-3(2,3)	4	1(1,2)-2(3)-3(3)
5	1(1)-2(2,3)-3(1)-4(2,3)	5	1(1,2)-2(3)-3(1,2)-4(3)
6	1(1)-2(2,3)-3(1)-4(1)-6(2,3)	6	1(1,2)-2(3)-3(1,2)-4(1,2)-6(3)
7	1(1)-2(1)-3(2,3)-4(2,3)	7	1(1,2)-2(1,2)-3(3)-4(3)
8	1(1)-2(1)-3(2,3)-4(1)-5(2,3)	8	1(1,2)-2(1,2)-3(3)-4(1,2)-5(3)

TABLE 10. Minimal cut sets of the system in MDD_1 & MDD_2.

number	cs_1	number	cs_2
1	1(2,3)	1	1(3)
2	4(2,3)	2	4(3)
3	2(2,3)-3(2,3)	3	2(3)-3(3)
4	2(2,3)-6(2,3)	4	2(3)-6(3)
5	3(2,3)-5(2,3)	5	3(3)-5(3)
6	5(2,3)-6(2,3)	6	5(3)-6(3)

combination algorithm, which need to get the failure or success state combination sets of components in water supply system.

V. CONCLUSION

We proposed a new MDD-GO method for multi-state reliability assessment and generated a complete MDD by implementing two mapping rules: first converting operators to MDD and later starting the conversion process of mapping the complete GO chart to the system MDD. The proposed method has the following advantages: (i) MDD can avoid the process of dealing with shared-signals and the correction of the probability calculation formula, consequently improving the calculation efficiency; (ii) through qualitative analysis, path and cut sets can be directly obtained based on the MDD-GO assessment, in contrast to models based on BNs; (iii) MDD is commonly used in a system with multi-state components that are widely used in the reliability field, and the mapping rules of generating MDD from the GO chart are concise and understandable; taken together, these findings indicate that MDD has good applied value in the reliability field. In this paper, the new MDD-GO method was determined to be advantageous through a case study: a two-branch water supply system.

As an exploration and extension of this research study, the size of the MDD converted from the GO chart has an obvious relationship with the order of the input variables that still requires further study. Furthermore, we will continue to consider an algorithm regarding the MDD-GO-FLOW model, because GO-FLOW is success-oriented analysis technique and basically has the same modeling approach and analysis procedure as the GO model. However, there are several discrepancies (e.g., basic conception and arithmetic) between these two models.

REFERENCES

[1] S. Zupei, G. Jia, and H. Xiangrui, "A new quantification algorithm for the GO methodology," *Rel. Eng. Syst. Safety.*, vol. 67, no. 3, pp. 241–247, Mar. 2000.

[2] B. B. Chu, "Go methodology: Overview manual," *Electr. Power Res. Inst.*, vol. 1, no. 1, pp. 125–130, 1983.

[3] M. A. A. Majid and M. N. Reshid, "Multi-state availability and reliability assessment of electric chillers of a district cooling plant," in *Proc. IEEE Colloq. Hum., Sci. Eng. (CHUSER)*, Mar. 2011, pp. 178–183.

[4] S. Daichman, I. Frenkel, L. Khvatskin, and A. Lisnianski, "On aging components impact on multi-state water cooling system: Lz-transform application for availability assessment," in *Proc. Int. Conf. Digit. Technol.*, Jul. 2013, pp. 156–161.

[5] T. Matsuoka and M. Kobayashi, "The GO-FLOW reliability analysis methodology—Analysis of common cause failures with uncertainty," *Nucl. Eng. Des.*, vol. 173, no. 3, pp. 205–214, 1997.

[6] R. E. Barlow and A. S. Wu, "Coherent systems with multi-state components," *Math. Oper. Res.*, vol. 3, no. 4, pp. 275–281, 1978.

[7] S. Zupei, G. Jia, and H. Xiangrui, "An exact algorithm dealing with shared signals in the GO methodology," *Rel. Eng. Syst. Safety*, vol. 73, no. 2, pp. 177–181, 2001.

[8] J. Xiuhong and D. Fuhai, "A new quantification algorithm with probability matrix in the GO methodology," *J. Inf. Comput. Sci.*, vol. 11, no. 14, pp. 5035–5042, 2014.

[9] J. Xiuhong and H. Ailing, "An improved GO methodology for system reliability analysis," *Mech. Sci., Electr. Eng. Comput.*, Dec. 2013, pp. 120–123.

[10] P. Weber and L. Jouffe, "Reliability modelling with dynamic Bayesian networks," *IFAC Proc. Volumes*, vol. 36, no. 5, pp. 57–62, Jun. 2003.

[11] H. Boudali and J. B. Duga, "A new Bayesian network approach to solve dynamic fault tree," in *Proc. Rel. Maintainability Symp.*, Jan. 2005, pp. 451–456.

[12] L.-L. Liu, Y. Ren, Z.-L. Wang, and D.-Z. Yang, "Algorithm based on Bayesian networks for GO methodology," *Syst. Eng. Electron.*, vol. 37, no. 1, pp. 212–218, Jan. 2015.

[13] D. Fan, Z. Wang L. Liu, and Y. Ren, "A modified GO-FLOW methodology with common cause failure based on discrete time Bayesian network," *Nucl. Eng. Des.*, vol. 305, no. 15, pp. 476–488, Aug. 2016.

[14] H. Langseth and L. Portinale, "Bayesian networks in reliability," *Rel. Eng. Syst. Safety.*, vol. 92, no. 1, pp. 92–108, Jan. 2007.

[15] T. P. K. Nguyen, J. Beugin, and J. Marais, "Method for evaluating an extended fault tree to analyse the dependability of complex systems: Application to a satellite-based railway system," *Rel. Eng. Syst. Safety*, vol. 133, pp. 300–313, Jan. 2015.

[16] F.-E. Majid and M.-A. Nezam, "An efficient algorithm for the multi-state two separate minimal paths reliability problem with budget constraint," *Rel. Eng. Syst. Safety.*, vol. 142, pp. 472–481, Oct. 2015.

[17] X. Zang, D. Wang, H. Sun, and K. S. Trivedi, "A BDD-based algorithm for analysis of multistate systems with multistate components," *IEEE Trans. Comput.*, vol. 52, no. 12, pp. 1608–1618, Dec. 2003.

[18] A. Shrestha, L. Xing, and Y. Dai, "Decision diagram based methods and complexity analysis for multi-state systems," *IEEE Trans. Rel.*, vol. 59, no. 1, pp. 145–161, Mar. 2010.

[19] D. M. Miller, "Multiple-valued logic design tools," in *Proc. 23rd Int. Symp. Multiple-Valued Logic*, May 1993, pp. 2–11.

[20] L. Xing and Y. S. Dai, "A new decision-diagram-based method for efficient analysis on multistate systems," *IEEE Trans. Depend. Sec. Comput.*, vol. 6, no. 3, pp. 161–174, Jul./Sep. 2009.

[21] M. Kvassay, E. Zaitseva, V. Levashenko, and J. Kostolny, "Minimal cut vectors and logical differential calculus," in *Proc. Int. Symp. Multiple-Valued Logic*, May 2014, pp. 167–172.

[22] M. Kvassay, E. Zaitseva, and V. Levashenko, "Minimal cut sets and direct partial logic derivatives in reliability analysis," in *Proc. Eur. Safety Rel. Conf. Safety Rel. Methodol. Appl. (ESREL)*, 2014, pp. 241–248.

[23] M. Kvassay, J. Rabcan, and P. Rusnak, "Multiple-valued logic in analysis of critical states of multi-state system," in *Proc. Int. Conf. Inf. Digit. Technol. (IDT)*, 2017, pp. 212–217.

[24] M. Kvassay and E. Zaitseva, "Topological analysis of multi-state systems based on direct partial logic derivatives," in *Recent Advances in Multi-State Systems Reliability* (Springer Series in Reliability Engineering). Cham, Switzerland: Springer, 2018, pp. 265–281.

[25] Y. Mo, L. Xing, S. V. Amari, and J. B. Dugan, "Efficient analysis of multi-state *k-out-of-n* systems," *Rel. Eng. Syst. Safety.*, vol. 133, pp. 95–105, Jan. 2015.

[26] Q. Zhai, L. Xing, R. Peng, and J. Yang, "Multi-valued decision diagram-based reliability analysis of *k-out-of-n* cold standby systems subject to scheduled backups," *IEEE Trans. Rel.*, vol. 64, no. 4, pp. 1310–1324, Dec. 2015.

[27] E. Zaitseva and V. Levashenko, "Multiple-valued logic mathematical approaches for multi-state system reliability analysis," *J. Appl. Logic*, vol. 11, no. 3, pp. 350–362, 2013.

[28] E. Zaitseva and S. Puuronen, "Representation and estimation of multi-state system reliability by decision diagrams," in *Proc. Joint ESREL SRA-Europe Conf. Safety, Rel. Risk Anal., Theory, Methods Appl.*, 2009, pp. 1995–2002.

[29] R. L. Williams and W. Y. Gateley, "Use of the GO methodology to directly generate minimal cut sets," in *Proc. Int. Conf. Nucl. Syst. Rel. Eng. Risk Assessment*, 1977, pp. 825–849.

[30] K. S. Corporation, R. L. Williams, and W. V. Gately, "GO methodology: Overview," Electr. Power Res. Inst., Palo Alto, CA, USA, Final Rep. TR: 78-016182, May 1978.

[31] L. Liu, D. Fan, and Y. Ren, "Enhanced GO methodology to support failure mode, effects and criticality analysis," *J. Intell. Manuf.*, vol. 2017, pp. 1–18, Jun. 2017.

[32] S. B. Akers, "Binary decision diagrams," *IEEE Trans. Comput.*, vol. C-27, no. 6, pp. 509–516, Jun. 1978.

[33] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Trans. Comput.*, vol. C-35, no. 8, pp. 677–691, Aug. 1986.

[34] A. B. Rauzy, J. Gauthier, and X. Leduc, "Assessment of large automatically generated fault trees by means of binary decision diagrams," *Proc. Inst. Mech. Eng., O, J. Risk Rel.*, vol. 221, no. 2, pp. 95–105, Jun. 2007.

[35] J. D. Murchland, "Fundamental concepts and relations for reliability analysis of multi-state systems and fault tree analysis," in *Proc. Theor. Appl. Aspects Syst.*, 1975, pp. 581–618.

[36] A. Lisnianski and G. Levitin, *Multi-State System Reliability: Assessment, Optimization and Applications*. Singapore: World Scientific, 2003.

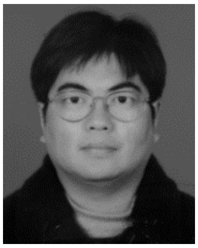
[37] T. Kam, T. Villa, and R. Brayton, *Synthesis of Finite State Machine: Function Optimization*. Norwell, MA, USA: Kluwer Academic., 1997.

[38] S. Li, S. Si, H. Dui, Z. Cai, and S. Sun, "A novel decision diagrams extension method," *Rel. Eng. Syst. Safety.*, vol. 126, pp. 107–115, Jun. 2014.

[39] Z. P. Shen and X. R. Huang, "Principle and application of GO methodology," in *A System Reliability Analysis Method*. Beijing, China: Tsinghua Univ. Press, 2014, pp. 1–114.

[40] Z. Tian, M. J. Zuo, and R. C. M. Yam, "The multi-state *k-out-of-n* systems and their performance evaluation," *IIE Trans.*, vol. 41, no. 1, pp. 32–44, Jun. 2008.

[41] G. Arulmozhi, "Exact equation and an algorithm for reliability evaluation of *K-out-of-N:G* system," *Rel. Eng. Syst. Safety*, vol. 78, no. 2, pp. 87–91, Nov. 2002.



YI REN (M'13) received the Ph.D. degree in reliability engineering and systems engineering from Beihang University, Beijing, China. He has over ten years of research and teaching experience in reliability engineering and system engineering. He is currently an Associate Professor, a Reliability Specialist, and a member of the Department of System Engineering of Engineering Technology, Faculty of System Engineering, Beihang University. He is also the Team Leader of the KW-GARMS, reliability engineering software platform. His recent research interests include the reliability of electronics, model-based reliability system engineering, computer-aided reliability, maintainability, and supportability design, and product life-cycle management. He received six Chinese ministry-level professional awards and one Chinese National-Level Professional Award.

CHENCHEN ZENG received the B.S. degree in industrial engineering from China Jiliang University, Hangzhou, China. She is currently pursuing the M.S. degree in reliability and systems engineering with the School of Reliability and Systems Engineering, Beihang University, Beijing, China. Her research interests include the modeling method of system reliability, and meta-synthesis of reliability, testability, and maintainability.

DONGMING FAN received the B.S. degree in detection guidance and control technology from Beihang University, Beijing, China, where he is currently pursuing the Ph.D. degree in reliability and systems engineering with the School of Reliability and Systems Engineering. His research interests include prognostics and health management, and reliability modeling and evaluation.

LINLIN LIU received the B.S. degree from the Faculty of Mathematics, Beihang University, Beijing, China, and the M.S. degree in weapon systems and utilization engineering and the Ph.D. degree in systems engineering from Beihang University, in 2009 and 2017, respectively, and the degree from the School of Reliability and Systems Engineering, Beihang University. His main research interests include model-based failure analysis and control, reliability modeling and prediction, and condition-based maintenance.

QIANG FENG (M'17) received the B.S. degree in mechanical engineering and the Ph.D. degree in reliability engineering and systems engineering from Beihang University, Beijing, China.

He is currently an Assistant Professor and a member of the Faculty of Systems Engineering, School of Reliability and Systems Engineering, Beihang University. He is a Supervisor of M.S. students, and also teaches two courses for undergraduate and master's students. He has published over 50 papers and two book chapters. He has led over ten projects supported by companies and industries. In the past few years, he has also participated in over ten projects supported by government and national commissions. His current research interests include the reliability modeling of complex system, maintenance planning, and the integrated design of product reliability and performance.