

# **EVMDD-Based Analysis and Diagnosis Methods of Multi-State Systems with Multi-State Components\***

SHINOBU NAGAYAMA<sup>1</sup>, TSUTOMU SASAO<sup>2</sup> AND JON T. BUTLER<sup>3</sup>

<sup>1</sup>*Dept. of Comp. and Network Eng., Hiroshima City Univ., Hiroshima, JAPAN*

<sup>2</sup>*Dept. of Comp. Science and Electronics, Kyushu Inst. of Tech., Iizuka, JAPAN*

<sup>3</sup>*Dept. of Electr. and Comp. Eng., Naval Postgraduate School, Monterey, CA USA*

*Received: June 18, 2012. Accepted: January 7, 2013.*

A multi-state system with multi-state components is a model of systems, where performance, capacity, or reliability levels of the systems are represented as states. It usually has more than two states, and thus can be considered as a multi-valued function, called a structure function. Since many structure functions are monotone increasing, their multi-state systems can be represented compactly by edge-valued multi-valued decision diagrams (EVMDDs). This paper presents an analysis method of multi-state systems with multi-state components using EVMDDs. Experimental results show that, by using EVMDDs, structure functions can be represented more compactly than existing methods using ordinary MDDs. Further, EVMDDs yield comparable computation time for system analysis. This paper also proposes a new diagnosis method using EVMDDs, and shows that the proposed method can infer the most probable causes for system failures more efficiently than conventional methods based on Bayesian networks.

*Keywords:* Multi-state systems with multi-state components; fault tolerant systems; structure functions; system analysis and diagnosis based on decision diagrams; EVMDDs.

## **1 INTRODUCTION**

Fault-tolerant techniques have been recently applied to various systems, such as computer servers, telecommunication equipment, water, gas, and electrical

---

\* A preliminary short version of the results of this paper was published at *the 42nd International Symposium on Multiple-Valued Logic* [14].

power distribution networks. These systems usually continue working with an acceptable or degraded performance level even if a fault occurs. Thus, unlike traditional systems that can be modeled by binary-state representations, these systems cannot be modeled by only two states: working and failure. In addition, with advances in technology, each component in a system also becomes fault tolerant. To model such a system, a multi-state system with multi-state components is often used [16, 20, 22].

Many fault-tolerance systems are designed by multiplexing components. However, multiplexing noncritical components to design a fault tolerant system is neither efficient nor cost effective. Also, if critical components are not sufficiently tolerant to faults, fault tolerance of the system is not sufficient. Thus, identifying which components are critical in achieving fault tolerance of the system is important, especially for safety-critical systems, such as flight control and nuclear power plant monitoring systems [2].

To identify critical components and system weaknesses, analyzing multi-state systems again and again by various assessment measures is required [16]. Among them, assessing the probability of each state of a multi-state system is essential to the design of a dependable fault tolerant system [20, 22]. Various methods to analyze multi-state systems efficiently have been proposed. Many existing methods are based on the Markov model [3]. However, they are impractical for a large multi-state system, because their time complexity is  $O(m^{3n})$ , where  $m$  is the number of states, and  $n$  is the number of components in a multi-state system [2]. To analyze large multi-state systems efficiently, methods based on binary decision diagrams (BDDs) [1, 2, 4, 22] and multi-valued decision diagrams (MDDs) [8, 15, 19, 20] have attracted much attention.

Since multi-state systems with multi-state components can be considered as multi-valued functions, called structure functions, they can be represented by BDDs and MDDs. Probabilities of states can be computed using BDDs and MDDs where the time complexity is proportional to the number of nodes in a decision diagram. BDDs represent structure functions by converting multi-valued variables and function values into binary vectors using one-hot encoding [22]. By using BDDs, various analysis methods that are well-established for binary-state systems can be directly applied to multi-state systems. However, converting to binary vectors produces many binary variables and many binary functions, and results in large BDDs. Thus, the use of MDDs is more natural and more promising for larger multi-state systems.

For recent large and complex multi-state systems, however, decision diagrams that represent systems more compactly are desired. Since structure functions are usually monotone increasing [19], they can be represented compactly using edge-valued MDDs (EVMDDs) [13]. However, analysis and diagnosis of multi-state systems using EVMDDs are not straightforward. As

far as we know, analysis and diagnosis methods using EVMDDs have never been reported. Thus, in this paper, we propose efficient analysis and diagnosis methods using EVMDDs.

This paper is organized as follows: Section 2 defines multi-state systems, and EVMDDs. Section 3 shows representations of structure functions using MDDs and EVMDDs, and in Section 4, we present an analysis method using EVMDDs. Experimental results for the analysis method are shown in Section 5. Section 6 proposes a new EVMDD-based diagnosis method. And, we make a few concluding remarks in Section 7.

## 2 PRELIMINARIES

This section defines multi-state systems, structure functions, and MDDs to represent structure functions.

### 2.1 Multi-State Systems and Structure Functions

**Definition 1.** A **multi-state system** is a model that represents the performance, capacity, or reliability levels of the systems as states. It usually has more than two states, and consists of one or more components that take multi-state as well.\*

**Definition 2.** We assume that a state of a multi-state system depends only on the states of components in the system. Then, a system with  $n$  components can be considered as a multi-valued function  $f(x_1, x_2, \dots, x_n) : R_1 \times R_2 \times \dots \times R_n \rightarrow M$ , where each  $x_i$  represents a component with  $r_i$  states,  $R_i = \{0, 1, \dots, r_i - 1\}$  is a set of the states, and  $M = \{0, 1, \dots, m - 1\}$  is a set of the  $m$  system states. This multi-valued function is called a **structure function** of the multi-state system.

**Definition 3.** A multi-valued function  $f(x_1, x_2, \dots, x_n)$  is a **monotone increasing function** iff for any  $x_i$ ,  $\alpha \leq \beta$ , where  $\alpha, \beta \in R_i$  implies

$$\begin{aligned} f(x_1, x_2, \dots, x_{i-1}, \alpha, x_{i+1}, \dots, x_n) \\ \leq f(x_1, x_2, \dots, x_{i-1}, \beta, x_{i+1}, \dots, x_n). \end{aligned}$$

In many applications, the states of a system and its components are totally ordered. That is, the deterioration of a component in the system deteriorates (or preserves) the whole system. Thus, structure functions are usually

---

\* It is also called a multi-state system with multi-state components.

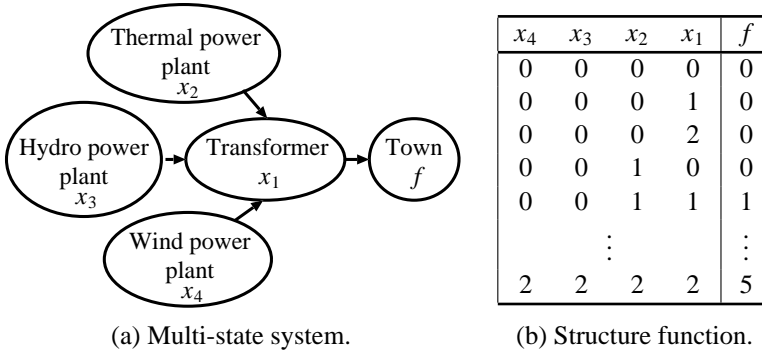


FIGURE 1

Multi-state system for an electrical power distribution and its structure function.

monotone increasing when a value is assigned to each state in ascending order (i.e. the worst state is 0 and the best state is  $m - 1$  or  $r_i - 1$ ).

**Example 1.** Figure 1(a) shows a multi-state system for an electrical power distribution system. In this figure, the transformer  $x_1$  and the power plants  $x_2, x_3, x_4$  have three states which correspond to supply levels: 0 (breakdown), 1 (partial supply), and 2 (full supply). And, the system has six states which correspond to the percentage of area of a town that is blacked out: 0 (complete blackout), 1 (90% blackout), 2 (60% blackout), 3 (30% blackout), 4 (10% blackout), and 5 (0% blackout).

In this way, by assigning a value to each state in ascending order, we obtain the 6-valued structure function  $f$  partially shown in Figure 1(b). This is a monotone increasing function. Note that a complete specification of this function can be determined by the function's representation as an MDD or EVMDD, as discussed in the next section.

## 2.2 Multi-Valued Decision Diagrams

**Definition 4.** A multi-valued decision diagram (MDD) is a rooted directed acyclic graph representing a multi-valued function  $f$ . The MDD is obtained by repeatedly applying the Shannon expansion to the multi-valued function [7]. It consists of non-terminal nodes representing sub-functions obtained from  $f$  by assigning values to certain variables. It also has terminal nodes representing function values. Each non-terminal node has multiple outgoing edges that correspond to the values of multi-valued variable. The MDD is ordered; i.e., the order of variables along any path from the root node to a terminal node is the same. When an MDD represents a function for

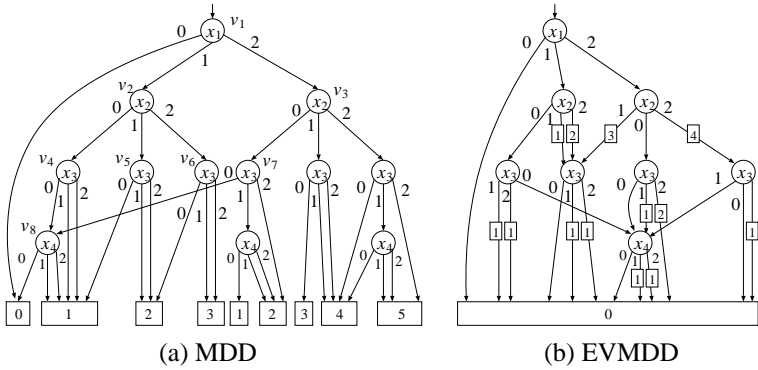


FIGURE 2  
MDD and EVMDD for the structure function.

which multi-valued variables have different domains, it is a heterogeneous MDD [11]. In the following, a heterogeneous MDD is also denoted simply as an MDD.

**Definition 5.** An **edge-valued MDD (EVMDD)** [13] is an extension of the MDD, and represents a multi-valued function. It consists of one terminal node representing 0 and non-terminal nodes with edges having integer weights; 0-edges always have zero weights. In an EVMDD, the function value is represented as the sum of weights for edges traversed from the root node to the terminal node.

**Example 2.** Figures 2(a) and (b) show an ordinary MDD and an EVMDD for the structure function of Example 1. For readability, some terminal nodes in the MDD are not combined.

### 3 MDDS AND EVMDDS FOR STRUCTURE FUNCTIONS

This section derives upper bounds on the number of nodes in an MDD and an EVMDD for a structure function. For simplicity, in the following theorems, we assume that all components  $x_i$  in a system have the same number  $r$  of states (i.e., all variables  $x_i$  have the same domain size). However, extending our results to a case where all variables  $x_i$  have different domain sizes is straightforward.

**Theorem 1.** For a structure function, the number of nodes in an MDD is at most

$$UB_M(m, n, r) = \frac{r^{n-l} - 1}{r - 1} + m^{r^l},$$

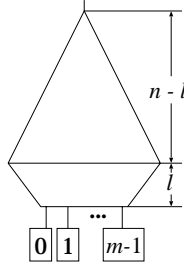


FIGURE 3  
Partition of an MDD.

where  $l$  is the largest integer satisfying  $r^{n-l} \geq m^l$ ,  $m$  is the number of system states,  $n$  is the number of components, and  $r$  is the number of states in all components.

*Proof.* Suppose that an MDD is partitioned into two parts: an upper part and a lower part, as shown in Figure 3. In this case, the lower part represents  $l$ -variable multi-valued functions, and the upper part represents the function that chooses one from them. The upper part has the maximum number of nodes when it forms a complete multi-valued tree. The number of nodes in the complete multi-valued tree is  $1 + r + r^2 + \dots + r^{n-l-1}$ . Thus, the maximum number of nodes in the upper part is

$$\frac{r^{n-l} - 1}{r - 1}. \quad (1)$$

The lower part has the maximum number of nodes when it represents all  $l$ -variable multi-valued functions. Since the total number of  $l$ -variable multi-valued functions is

$$m^{r^l}, \quad (2)$$

this is the maximum number of nodes in the lower part including terminal nodes. From (1) and (2), the number of nodes in the MDD is at most

$$\frac{r^{n-l} - 1}{r - 1} + m^{r^l}.$$

The number of multi-valued functions that can be represented in the lower part does not exceed the number of functions that can be chosen by the upper part:  $r^{n-l}$ . Therefore, we have the relation  $r^{n-l} \geq m^{r^l}$ . ■

Theorem 1 shows that an upper bound for an MDD depends only on  $m$ ,  $n$ , and  $r$ . It is independent of monotonicity of structure functions. Although this is an upper bound over all MDDs, it is a tight upper bound for monotone increasing functions, since multi-terminal decision diagrams cannot represent monotonicity of functions compactly [12]. In many applications, structure functions are usually monotone increasing. Thus, decision diagrams suitable for monotone functions are preferable. Since EVMDDs can represent monotone functions compactly, EVMDDs are preferable for many monotone structure functions.

**Definition 6.** Let  $\mathbb{N}_0$  be the set of nonnegative integers, and let  $p \in \mathbb{N}_0$ . An integer function  $f(X) : \mathbb{N}_0 \rightarrow \mathbb{N}_0$  such that  $0 \leq f(X + 1) - f(X) \leq p$  and  $f(0) = 0$  is an **Mp-monotone increasing function** on  $\mathbb{N}_0$ . That is, an Mp-monotone increasing function  $f(X)$  satisfies  $f(0) = 0$ , and increasing  $X$  by one increases the value of  $f(X)$  by at most  $p$ .  $p$  is called the increment value.

A monotone multi-valued function can be converted into an Mp-monotone increasing function by considering the set of multi-valued variables  $x_i$  as an  $r$ -valued vector:

$$X = (x_n, x_{n-1}, \dots, x_1)_r,$$

and EVMDDs for monotone multi-valued functions have the same complexity as EVMDDs for Mp-monotone increasing functions [13]. In this paper, Mp-monotone increasing functions obtained in this way from  $n$ -variable multi-valued functions are called  $n$ -variable Mp-monotone increasing functions.

In the following, we derive an upper bound of an EVMDD for an  $n$ -variable Mp-monotone increasing function. To derive an upper bound, we begin by defining a  $(p + 1)$ -valued 0-preserving function, and show a lemma on the number of distinct  $n$ -variable Mp-monotone increasing functions.

**Definition 7.** An  $n$ -variable multi-valued function  $h : \{0, 1, \dots, r - 1\}^n \rightarrow \{0, 1, \dots, p\}$  such that  $h(0, 0, \dots, 0) = 0$  is an  $n$ -variable  $(p + 1)$ -valued 0-preserving function. This is an extension of the 0-preserving function for logic function [18].

**Lemma 1.** The number of distinct  $n$ -variable Mp-monotone increasing functions is  $(p + 1)^{n-1}$ .

*Proof.* Let  $h(Y)$  be an  $n$ -variable  $(p + 1)$ -valued 0-preserving function, where  $Y = (y_{n-1} \ y_{n-2} \ \dots \ y_0)_r$ . For each  $h(Y)$ , there exists an  $n$ -variable

$Mp$ -monotone increasing function

$$f(X) = \sum_{Y=0}^X h(Y).$$

Conversely, for any given  $n$ -variable  $Mp$ -monotone increasing function  $f(X)$ , there exists a  $(p+1)$ -valued 0-preserving function. The number of different  $h$ 's is  $(p+1)^{r^{n-1}}$ . Therefore, we have the lemma. ■

**Theorem 2.** *For an  $n$ -variable  $Mp$ -monotone increasing function, the number of nodes in an EVMDD is at most*

$$UB_E(n, p, r) = \frac{r^{n-l} - 1}{r - 1} + \sum_{i=0}^l (p+1)^{r^i - 1} - l,$$

where  $l$  is the largest integer satisfying  $r^{n-l} \geq (p+1)^{r^l - 1}$ ,  $p$  is the increment value, and  $r$  is the number of component states.

*Proof.* Suppose that an EVMDD is partitioned into an upper part and a lower part, as shown in Figure 3. In this case, the lower part represents  $Mp$ -monotone increasing functions having fewer than or equal to  $l$  variables, and the upper part represents the function that chooses one from them. Since the upper part has the maximum number of nodes when it forms a complete  $r$ -valued tree, its maximum number of nodes is the same as (1).

The lower part has the maximum number of nodes when it represents all  $Mp$ -monotone increasing functions with fewer than or equal to  $l$  variables. From Lemma 1, the total number of functions except for the zero constant function is

$$\sum_{i=1}^l ((p+1)^{r^i - 1} - 1) = \sum_{i=1}^l (p+1)^{r^i - 1} - l \quad (3)$$

This is the maximum number of nodes in the lower part excluding the terminal node. Thus, by summing (1), (3), and 1 (for the terminal node), we have the theorem. The number of  $Mp$ -monotone increasing functions that can be represented in the lower part does not exceed the number of functions that can be chosen by the upper part. Therefore, we have the relation  $r^{n-l} \geq (p+1)^{r^l - 1}$ . ■



Theorem 2 shows that the upper bound for an EVMDD depends on the value of  $p$ , not the number of system states  $m$ . Thus, even if  $m$  is large, EVMDDs have a small number of nodes when the value of  $p$  is small. In the future, systems will become more complex, and thus,  $m$  will become larger. For such systems, MDDs require many nodes. On the other hand, EVMDDs can represent such systems compactly if the value of  $p$  is small.

#### 4 ANALYSIS METHODS USING MDDS AND EVMDDS

This section formulates a problem of system analysis, and then presents an algorithm to solve the problem using EVMDDs.

**Definition 8.** *The probability that a structure function  $f$  has the value  $s$  is denoted by  $P_s(f = s)$ , where  $s \in \{0, 1, \dots, m - 1\}$ . The probability that a component  $x_i$  has the value  $c$  is denoted by  $P_c(x_i = c)$ , where  $c \in \{0, 1, \dots, r_i - 1\}$ .*

**Problem 1.** *Given a structure function  $f$  of a multi-state system and the probability of each state of each component in the system  $P_c(x_i = c)$ , compute the probability of each state of the multi-state system  $P_s(f = s)$ .*

In this problem, we assume that the probabilities of all component states are independent of each other.

##### 4.1 Analysis Method Using MDDs

Problem 1 can be solved efficiently using *node traversing probabilities* in an MDD that compute the average path length in an MDD [10].

**Definition 9.** *In an MDD, a sequence of edges and nodes leading from the root node to a terminal node is a **path**. The **node traversing probability**, denoted by  $NTP(v_i)$ , is the probability that an assignment of values to variables selects a path that includes the node  $v_i$ .*

Since terminal nodes of an MDD for a structure function represent system states, node traversing probabilities of terminal nodes correspond to the probabilities of system states. Node traversing probabilities can be computed by visiting each node only once in breadth-first order starting from the root node. Thus, the time complexity of this analysis method is  $O(N_M)$ , where  $N_M$  is the number of nodes in an MDD. Other existing methods whose time complexity is  $O(N_M)$  also analyze multi-state systems in a similar way [8, 19, 20].

**Example 3.** Let us compute node traversing probabilities for the MDD in Figure 2(a). In this example, we assume that all states of each component occur with the same probability,  $1/3$ .

First, we have  $NTP(v_1) = 1$  for the root node  $v_1$ , since the root node occurs in all paths. Then, we compute  $NTP(v_2) = NTP(v_1) \times 1/3$  and  $NTP(v_3) = NTP(v_1) \times 1/3$  in a breadth-first order. Similarly, by computing NTPs in a top-down manner, we have  $NTP(v_4) = NTP(v_2)/3$ ,  $NTP(v_5) = NTP(v_2)/3$ ,  $NTP(v_6) = NTP(v_2)/3$ , and  $NTP(v_7) = NTP(v_3)/3$ . Since at a re-convergence node  $v_8$ , all NTPs received from parent nodes are summed up, we have  $NTP(v_8) = NTP(v_4)/3 + NTP(v_7)/3$ . Finally, we have the node traversing probabilities of terminal nodes:  $NTP(0) = 29/81$ ,  $NTP(1) = 14/81$ ,  $NTP(2) = 14/81$ ,  $NTP(3) = 1/9$ ,  $NTP(4) = 10/81$ , and  $NTP(5) = 5/81$ .

This shows that, in this electrical power distribution system, the town becomes completely blacked out with probability 36% ( $NTP(0) = 29/81$ ).

## 4.2 Analysis Method Using EVMDDs

In an EVMDD, a function value is represented by a sum of edge values, rather than a terminal node. Thus, we cannot solve Problem 1 using only node traversing probabilities, and another analysis method is needed. In this subsection, we present a bottom-up approach that computes probabilities for each sub-function incrementally.

Figure 4 shows the proposed analysis algorithm. This algorithm visits each node only once in depth-first order starting from the root node, and analyzes a sub-function represented by each node recursively. Probabilities for a

Input:	An EVMDD for a structure function of a multi-state system, and the probability of each state of each component in the system $P_c(x_i = c)$ .
Output:	Probability of each state of the multi-state system $P_s(f = s)$ .
Step:	The following procedures are applied to each node recursively from the root node.
1.	If a node $v$ has been already visited, then return the probabilities for the sub-function $f_v$ that have been already computed. Else, go to the next step.
2.	If a node $v$ is the terminal node $T$ , then return the probability for the constant zero function: $P_s(f_T = 0) = 1$ . Else, go to the next step.
3.	Visit all child nodes $u_j$ of $v$ , and obtain the probabilities for the sub-functions $f_{u_j}$ represented by $u_j$ .
4.	Multiply the obtained probabilities for a sub-function $P_s(f_{u_j} = s)$ by the probability that the component $x_i$ selects the sub-function $P_c(x_i = c)$ .
5.	Each function value $f_{u_j} = s$ at each child node $u_j$ becomes a function value $f_v = s + e_j$ at the node $v$ because of its edge value $e_j$ . Thus, the probabilities $P_s(f_{u_j} = s) \times P_c(x_i = c)$ obtained by the step 4 are added to $P_s(f_v = s + e_j)$ , and they are summed up (merged) in each function value at $v$ .
6.	Return the merged probabilities to the parent node.

FIGURE 4  
Proposed analysis algorithm using EVMDDs.

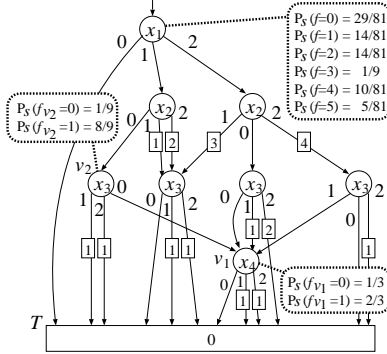


FIGURE 5  
Analysis of a multi-state system using EVMDD.

function represented by a node can be computed by merging probabilities for sub-functions represented by its child nodes. Thus, the algorithm shown in Figure 4 efficiently computes the probability of each state of a multi-state system. Since the algorithm visits each node only once, its time complexity is  $O(N_E)$ , where  $N_E$  is the number of nodes in an EVMDD.

**Example 4.** Let us compute the probability of each state of the multi-state system using the EVMDD in Figure 5. As with the previous example, we assume that all states of each component occur with the same probability,  $1/3$ .

First, we have  $P_s(f_T = 0) = 1$  at the terminal node  $T$ . Then, we compute the probability for a sub-function at the node  $v_1$ . Since this node has two edges pointing to  $T$  whose values are 1, and the two edges represent  $f_{v_1} = 1$ , we have

$$\begin{aligned} P_s(f_T = 0) \times P_c(x_4 = 1) &= 1/3, \\ P_s(f_T = 0) \times P_c(x_4 = 2) &= 1/3, \text{ and thus,} \\ P_s(f_{v_1} = 1) &= P_s(f_T = 0) \times P_c(x_4 = 1) \\ &\quad + P_s(f_T = 0) \times P_c(x_4 = 2) = 2/3. \end{aligned}$$

Thus, we have  $P_s(f_{v_1} = 0) = 1/3$  and  $P_s(f_{v_1} = 1) = 2/3$  for  $v_1$ , as shown in Figure 5. Since node  $v_2$  also has two edges pointing to  $T$  whose values are 1, the same probabilities are obtained from  $T$ . Since the 0-edge of  $v_2$  points to  $v_1$ , the probabilities at  $v_1$  are multiplied by  $1/3$ , and they are added to the probabilities from  $T$ . Thus, we have  $P_s(f_{v_2} = 0) = 1/9$  and  $P_s(f_{v_2} = 1) = 8/9$ . Similarly, by performing the same computation at each node in the depth-first order, we have the following at the root node:  $P_s(f = 0)$

$= 29/81$ ,  $P_s(f = 1) = 14/81$ ,  $P_s(f = 2) = 14/81$ ,  $P_s(f = 3) = 1/9$ ,  $P_s(f = 4) = 10/81$ , and  $P_s(f = 5) = 5/81$ . Note that these are consistent with the results obtained by MDDs in Example 3.

When a structure function is monotone increasing, the number of nodes in an EVMDD  $N_E$  is smaller than for non-monotone increasing functions, and thus, computation time is shorter. Of course, the proposed method can be applied to nonmonotonic structure functions used in some applications [20, 22] as well.

## 5 EXPERIMENTAL RESULTS

To show the effectiveness of the proposed analysis method, we compare it with a conventional analysis method based on MDDs in terms of their size and runtime using various structure functions. To show the difference between the two methods clearly, we need large structure functions since both the analysis methods are compact and fast. Unfortunately, however, benchmark functions of such sizes were not found in the literature. Since structure functions are usually monotone increasing, we randomly generated M1-monotone increasing functions, and used them as structure functions for experiments in this paper. The analysis algorithms based on MDDs and EVMDDs are implemented using the following computer environment: CPU: Intel Core2 Quad Q6600 2.4GHz, memory: 4GB, OS: CentOS 5.7, and C-compiler: gcc -O3 (version 4.1.2). Table 1 shows the experimental results for randomly generated  $m$ -state systems with  $n$  components, each component having three states.

From this table, we can see that EVMDDs have fewer nodes than MDDs for all the functions. Especially, as the number of states  $m$  becomes larger, EVMDDs are much smaller than MDDs. We expect that systems will become more complex in the future, and that  $m$ , the number of system states, will become larger. Thus, EVMDDs whose size is independent of the number of states are more promising. However, when  $m$  is very small, MDDs are faster, since sizes of MDDs are small enough. In Table 1, when  $m = 3$ , only two terminal nodes are reduced in EVMDDs. Thus, using EVMDDs for such systems is not effective.

As for the computation time, the proposed method using EVMDDs is comparable to the conventional method using MDDs. Therefore, we can say that EVMDDs are suitable for compact representation and efficient analysis of *many*-state systems.

Figure 6 compares MDDs and EVMDDs for 15-variable Mp-monotone increasing functions using various values of  $p$ , in terms of the number of

<i>n</i>	<i>m</i>	Number of nodes			Computation time ( $\mu$ sec.)		
		MDD	EVMDD	Ratio	MDD	EVMDD	Ratio
5	3	12	10	83%	0.30	0.96	316%
5	10	36	18	50%	1.00	2.39	240%
10	3	17	15	88%	0.60	1.61	271%
10	10	77	57	74%	2.71	6.87	253%
10	100	599	265	44%	24.28	40.54	167%
10	1,000	4,201	907	22%	216.60	250.31	116%
15	3	32	30	94%	1.31	3.03	232%
15	10	120	105	88%	4.54	12.14	267%
15	100	1,098	708	64%	57.82	94.26	163%
15	1,000	9,010	3,362	37%	529.38	630.49	119%
15	10,000	70,140	11,474	16%	4,357.00	3,800.00	87%
15	100,000	495,224	62,759	13%	60,861.00	52,955.00	87%

*n*: Number of 3-state components.

*m*: Number of system states.

Ratio:  $EVMDD / MDD \times 100$  (%)

The computation time is an average time obtained by running the same computation 1, 000, 000 times, and dividing its total time by 1, 000, 000.

TABLE 1

MDDs and EVMDDs for *m*-state systems with *n* 3-state components.

nodes and analysis time. Figure 6 shows that the number of nodes and analysis time in MDDs are independent of the value of *p*, as described in Section 3. On the other hand, the number of nodes and analysis time in EVMDDs increase as the value of *p* increases. However, its rate of increase is low. In this example, the number of nodes increases by up to 16%, and the analysis time increases by up to 11% for each increment of *p*. Therefore, we can say that EVMDDs are efficient when *p* is small.

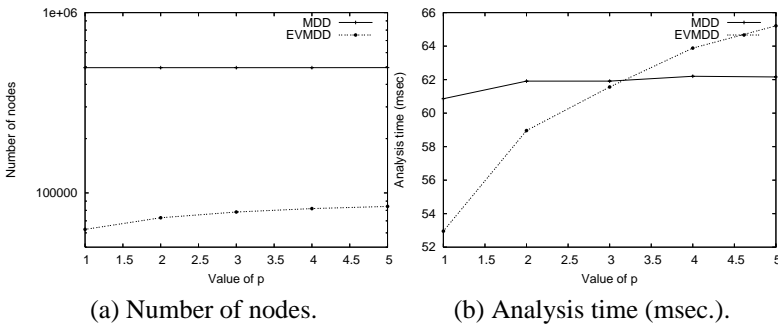


FIGURE 6

MDDs and EVMDDs for 15-variable *Mp*-monotone increasing functions.

## 6 DIAGNOSIS METHOD USING EVMDDS

This section formulates a problem of system diagnosis, and then proposes an EVMDD-based method to infer the most probable causes for system failures using probabilities of systems obtained by the analysis method in Figure 4. In this paper, we infer such causes by computing conditional probabilities (also known as posterior probabilities).

**Definition 10.** *A conditional probability, denoted by  $P(A|B)$ , is the probability that an event  $A$  occurs given that an event  $B$  has occurred.*

By considering event  $B$  as a system state, and event  $A$  as a component state, the probability that the component state causes the system state is obtained. Then, we infer the most probable causes for the system state from the probability. Thus, we formulate a problem of system diagnosis as follows:

**Problem 2.** *Given a structure function  $f$ , the probability of each state of each component  $P_c(x_i = c)$ , a system state  $f = s$ , and a component state  $x_i = c$ , compute the conditional probability  $P(x_i = c|f = s)$ .*

The conditional probability can be computed by Bayes' theorem:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}.$$

Bayesian networks infer causes of various events based on this theorem [5,6].

Since in the diagnosis of multi-state systems,  $A$  is a component state (i.e.,  $x_i = c$ ) and  $B$  is a system state (i.e.,  $f = s$ ), Bayes' theorem is rewritten as follows:

$$P(x_i = c|f = s) = \frac{P(f = s|x_i = c) \times P_c(x_i = c)}{P_s(f = s)}. \quad (4)$$

From this equation, we can solve Problem 2, if the probability  $P(f = s|x_i = c)$  is known. This is because  $P_c(x_i = c)$  is given, and  $P_s(f = s)$  can be obtained by the analysis algorithm in Figure 4. Thus, in this section, we propose an efficient algorithm to compute the probability  $P(f = s|x_i = c)$ .

### 6.1 Algorithm to Compute Conditional Probability Using EVMDD

From the definition of the conditional probability, the probability  $P(f = s|x_i = c)$  is equivalent to the probability of the system state  $P_s(f = s)$  when  $x_i = c$ . This can be obtained by taking the cofactor of  $f$  with respect to

Input:	An EVMDD for a structure function of a multi-state system, the probability of each component state in the system $P_c$ , a system state $f = s$ , and a component state $x_i = c$ .
Output:	The conditional probability $P(f = s x_i = c)$ .
Step:	<p>The following procedures are applied to each node recursively from the root node, and the probability <math>P(f = s x_i = c)</math> at the root node is the solution.</p> <ol style="list-style-type: none"> <li>1. If a node <math>v</math> has been already visited, then return the probabilities for the sub-function <math>f_v</math> that have been already computed. Else, go to the next step.</li> <li>2. If a node <math>v</math> is the terminal node <math>T</math>, then return the probability for the constant zero function: <math>P(f_T = 0 x_i = c) = 1</math>. Else, go to the next step.</li> <li>3. If the variable assigned to <math>v</math> is <math>x_i</math>, then visit a child node <math>u_c</math> connected to the edge of <math>x_i = c</math>. Else, visit all child nodes <math>u_j</math>. Obtain the probabilities for the visited sub-functions. <u>If the variable is <math>x_i</math>, then go to the step 5. Else, go to the next step.</u></li> <li>4. Multiply the obtained probabilities for a sub-function by the probability that the variable selects the sub-function <math>P_c</math>.</li> <li>5. Each function value <math>s_u</math> at each visited child node becomes a function value <math>f_v = s_u + e_j</math> at the node <math>v</math> because of its edge value <math>e_j</math>. Thus, the probabilities for <math>s_u</math> are added to <math>P(f_v = s_u + e_j x_i = c)</math>, and they are summed up in each function value at <math>v</math>.</li> <li>6. Return the merged probabilities to the parent node.</li> </ol>

FIGURE 7  
Algorithm to compute conditional probabilities using EVMDD

$x_i = c$  [18], and computing the probability that the cofactor has  $s$ . Since cofactors can be computed efficiently by using decision diagrams, we can compute the probability  $P(f = s|x_i = c)$  efficiently by applying an algorithm similar to Figure 4 to a cofactor.

Figure 7 shows the proposed algorithm to compute the conditional probability  $P(f = s|x_i = c)$  using an EVMDD. Actually, this algorithm computes the conditional probabilities for all system states  $P(f = 0|x_i = c)$ ,  $P(f = 1|x_i = c)$ ,  $\dots$ ,  $P(f = m - 1|x_i = c)$  simultaneously. Thus, we can diagnose effects of the component state ( $x_i = c$ ) to all the system states efficiently using this algorithm.

This algorithm is based on the analysis algorithm in Figure 4 with the underlined parts shown in Figure 7 that compute a cofactor added to the analysis algorithm. Therefore, its time complexity is the same as the analysis algorithm,  $O(N_E)$ , where  $N_E$  is the number of nodes in an EVMDD.

**Example 5.** Let us compute the conditional probability  $P(f = 0|x_3 = 0)$  using the EVMDD in Figure 8. As with the previous examples, we assume that all states of each component occur with the same probability,  $1/3$ .

First, we have  $P(f_T = 0|x_3 = 0) = 1$  at the terminal node  $T$ . As shown in Example 4, we compute probabilities at the node  $v_1$ , yielding  $P(f_{v_1} = 0|x_3 = 0) = 1/3$  and  $P(f_{v_1} = 1|x_3 = 0) = 2/3$ . Then, we compute probabilities at the node  $v_2$ . Since the variable assigned to  $v_2$  is  $x_3$ , we visit only a child node connected to the 0-edge. The child node is  $v_1$ , and thus, the probabilities at  $v_2$  are the same as the probabilities at  $v_1$ . Similarly, by performing the same

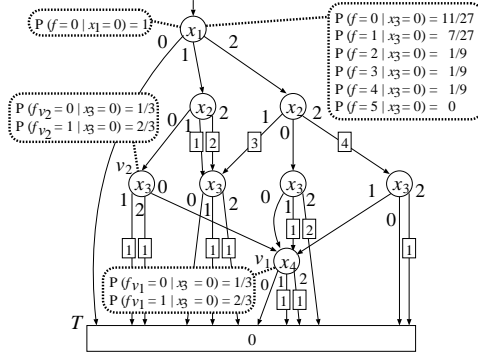


FIGURE 8  
Computing conditional probabilities using EVMD.

computation as the analysis method at each node in the depth-first order, we have the following probabilities at the root node:  $P_s(f = 0|x_3 = 0) = 11/27$ ,  $P_s(f = 1|x_3 = 0) = 7/27$ ,  $P_s(f = 2|x_3 = 0) = 1/9$ ,  $P_s(f = 3|x_3 = 0) = 1/9$ ,  $P_s(f = 4|x_3 = 0) = 1/9$ , and  $P_s(f = 5|x_3 = 0) = 0$ . Therefore,  $P(f = 0|x_3 = 0) = 11/27$ , and thus, by the equation (4),

$$P(x_3 = 0|f = 0) = \frac{P(f = 0|x_3 = 0) \times P_c(x_3 = 0)}{P_s(f = 0)} = \frac{11}{29} \approx 0.379.$$

This implies that the breakdown of the hydro power plant causes a complete blackout of the town with probability 38%.

On the other hand,  $P(f = 0|x_1 = 0) = 1$  is readily obtained because the 0-edge of the root node points to the terminal node. Thus, we have

$$P(x_1 = 0|f = 0) = \frac{P(f = 0|x_1 = 0) \times P_c(x_1 = 0)}{P_s(f = 0)} = \frac{27}{29} \approx 0.931.$$

This implies that the breakdown of the transformer causes a complete blackout of the town with probability 93%. Since this is the most probable cause for a complete blackout of the town, the transformer turns out to be a critical component in achieving fault tolerance of this system. It agrees with our intuition.

## 6.2 Comparison with Bayesian Networks

Bayesian networks represent cause-and-effect relationships among components and a system, and compute the probability of each system state  $P_s(f)$



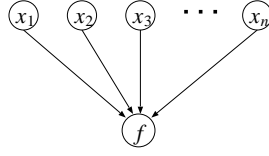


FIGURE 9  
Bayesian networks representing systems with independent components.

=  $s$ ) as follows [23]:

$$P_s(f = s) = \sum_{\vec{c} \in R^n} P(f = s, x_1, x_2, \dots, x_n), \tag{5}$$

where  $R^n$  is a set of value assignments to all the variables  $x_i$ , and  $P(f = s, x_1, x_2, \dots, x_n)$  is a joint probability of  $f = s$  and  $(x_1, x_2, \dots, x_n) = \vec{c} = (c_1, c_2, \dots, c_n)$ . Since, in the systems considered in this paper, all components are independent of each other, their Bayesian networks become like that of Figure 9, and

$$\begin{aligned} P(f = s, x_1 = c_1, x_2 = c_2, \dots, x_n = c_n) = \\ P(f = s | x_1 = c_1, x_2 = c_2, \dots, x_n = c_n) \\ \times P_c(x_1 = c_1) \times P_c(x_2 = c_2) \times \dots \times P_c(x_n = c_n). \end{aligned}$$

The conditional probability  $P(f = s | x_1 = c_1, x_2 = c_2, \dots, x_n = c_n)$  is

$$P(f = s | x_1 = c_1, x_2 = c_2, \dots, x_n = c_n) = \begin{cases} 0 & (f(c_1, c_2, \dots, c_n) \neq s) \\ 1 & (f(c_1, c_2, \dots, c_n) = s) \end{cases},$$

because  $f$  is a function of  $x_1, x_2, \dots, x_n$ . Therefore, (5) is obtained by a sum of probabilities that an input vector  $\vec{c}$  satisfies  $f = s$ . Its time complexity is  $O(r^n)$ . This is obviously inefficient. Also, in a diagnosis using Bayesian networks, one can assume that values of  $P(f = s | x_i = c)$  are given to compute (4).

On the other hand, our diagnosis method using EVMDDs can compute (4) even if the values of  $P(f = s | x_i = c)$  are not given, as shown in the previous subsection. Since the time complexities to compute  $P_s(f = s)$  and  $P(f = s | x_i = c)$  are both  $O(N_E)$ , our methods using EVMDDs analyze and diagnose multi-state systems more efficiently than conventional methods based on Bayesian networks. In fact, our method using EVMDDs analyzes the largest 15-component system in Table 1 about 12.6 times faster than a

conventional method based on Bayesian networks. The conventional method requires 667.4 msec to analyze the same system.

Although this paper assumes that all components are independent of each other, an MDD-based method can efficiently analyze multi-state systems in which components have interdependent states as well [9].

## 7 CONCLUSION AND COMMENTS

This paper presents an efficient analysis method of multi-state systems using EVMDDs. It is somewhat more complicated than existing methods using ordinary MDDs because a state of the system is represented by a sum of edge values. However, the computation time of our EVMDD-based analysis method is comparable to methods using MDDs, since the time complexity is asymptotically proportional to the number of nodes in an EVMDD, and EVMDDs have fewer nodes than MDDs. Especially, for systems with many states, our analysis method is effective because EVMDDs are much smaller than MDDs.

This paper focuses only on monotone increasing structure functions to emphasize the effectiveness of EVMDDs. However, there exist applications using nonmonotonic structure functions [20, 22]. Even for such structure functions, EVMDDs are not larger than MDDs [12]. Thus, our analysis method is effective for a wide range of structure functions.

We also propose an efficient diagnosis method using EVMDDs. The proposed diagnosis method can infer the most probable causes for system failures more efficiently than conventional methods based on Bayesian networks.

In this paper, we used randomly generated M1-monotone increasing functions for our experiments, since benchmarks of multi-state systems were unavailable. However, there could be functions more suitable for multi-state systems. Thus, we will study such functions. We will also study how to generate EVMDDs directly from multi-state systems without using MDDs for structure functions.

## ACKNOWLEDGMENTS

This research is partly supported by the Grant in Aid for Scientific Research of the Japan Society for the Promotion of Science (JSPS), funds from Ministry of Education, Culture, Sports, Science, and Technology (MEXT), the MEXT Grant-in-Aid for Scientific Research (C), (No. 22500050), 2012, and Hiroshima City University Grant for Special Academic Research (General Studies), (No. 0206), 2012.

We would like to thank Prof. Dan A. Simovici for a discussion on Bayesian networks. The reviewers' comments were helpful in improving the paper.

## REFERENCES

- [1] J. D. Andrews and S. J. Dunnett, "Event-tree analysis using binary decision diagrams," *IEEE Transactions on Reliability*, Vol. 49, No. 2, pp. 230–238, June 2000.
- [2] Y.-R. Chang, S. V. Amari, and S.-Y. Kuo, "Reliability evaluation of multi-state systems subject to imperfect coverage using OBDD," *Proc. of the 2002 Pacific Rim International Symposium on Dependable Computing (PRDC'02)*, pp. 193–200, 2002.
- [3] S. A. Doyle, J. B. Dugan, and F. A. Patterson-Hine, "A combinatorial approach to modeling imperfect coverage," *IEEE Transactions on Reliability*, Vol. 44, No. 1, pp. 87–94, Mar. 1995.
- [4] S. A. Doyle and J. B. Dugan, "Dependability assessment using binary decision diagrams (BDDs)," *25th International Symposium on Fault-Tolerant Computing (FTCS)*, pp. 249–258, June 1995.
- [5] F. V. Jensen, *An Introduction to Bayesian Networks*, University College London (UCL) Press, 1996.
- [6] F. V. Jensen, *Bayesian Networks and Decision Graphs*, Springer-Verlag New York Inc., 2001.
- [7] T. Kam, T. Villa, R. K. Brayton, and A. L. Sangiovanni-Vincentelli, "Multi-valued decision diagrams: Theory and applications," *Multiple-Valued Logic: An International Journal*, Vol. 4, No. 1-2, pp. 9–62, 1998.
- [8] T. W. Manikas, M. A. Thornton, and D. Y. Feinstein, "Using multiple-valued logic decision diagrams to model system threat probabilities," *41st International Symposium on Multiple-Valued Logic*, pp. 263–267, May 2011.
- [9] T. W. Manikas, D. Y. Feinstein, and M. A. Thornton, "Modeling medical system threats with conditional probabilities using multiple-valued logic decision diagrams," *42nd International Symposium on Multiple-Valued Logic*, pp. 244–249, May 2012.
- [10] S. Nagayama A. Mishchenko, T. Sasao, and J. T. Butler, "Exact and heuristic minimization of the average path length in decision diagrams," *Journal of Multiple-Valued Logic and Soft Computing*, Vol. 11, No. 5-6, pp. 437–465, Aug. 2005.
- [11] S. Nagayama and T. Sasao, "On the optimization of heterogeneous MDDs," *IEEE Trans. on CAD*, Vol. 24, No. 11, pp. 1645–1659, Nov. 2005.
- [12] S. Nagayama and T. Sasao, "Complexities of graph-based representations for elementary functions," *IEEE Trans. on Computers*, Vol. 58, No. 1, pp. 106–119, Jan. 2009.
- [13] S. Nagayama, T. Sasao, and J. T. Butler, "A systematic design method for two-variable numeric function generators using multiple-valued decision diagrams," *IEICE Trans. on Information and Systems*, Vol. E93-D, No. 8, pp. 2059–2067, Aug. 2010.
- [14] S. Nagayama, T. Sasao, and J. T. Butler, "Analysis of multi-state systems with multi-state components using EVMDDs," *42nd International Symposium on Multiple-Valued Logic*, pp. 122–127, May, 2012.
- [15] P. Ongsakorn, K. Turney, M. Thornton, S. Nair, S. Szygenda, and T. Manikas, "Cyber threat trees for large system threat cataloging and analysis," *4th IEEE International Systems Conference*, pp. 610–615, April, 2010.

- [16] J. E. Ramirez-Marquez and D. W. Coit, "Composite importance measures for multi-state systems with multi-state components," *IEEE Transactions on Reliability*, Vol. 54, No. 3, pp. 517–529, Sept. 2005.
- [17] T. Sasao and M. Fujita (eds.), *Representations of Discrete Functions*, Kluwer Academic Publishers 1996.
- [18] T. Sasao, *Switching Theory for Logic Synthesis*, Kluwer Academic Publishers 1999.
- [19] L. Xing and J. B. Dugan, "Dependability analysis using multiple-valued decision diagrams," *Proc. of 6th International Conference on Probabilistic Safety Assessment and Management*, June 2002.
- [20] L. Xing and Y. Dai, "A new decision-diagram-based method for efficient analysis on multistate systems," *IEEE Transactions on Dependable and Secure Computing*, Vol. 6, No. 3, pp. 161–174, 2009.
- [21] S. N. Yanushkevich, D. M. Miller, V. P. Shmerko, and R. S. Stankovic, *Decision Diagram Techniques for Micro- and Nanoelectronic Design*, CRC Press, Taylor & Francis Group, 2006.
- [22] X. Zang, D. Wang, H. Sun, and K. S. Trivedi, "A BDD-based algorithm for analysis of multistate systems with multistate components," *IEEE Transactions on Computers*, Vol. 52, No. 12, pp. 1608–1618, Dec. 2003.
- [23] Z. Zhou, G. Jin, D. Dong, and J. Zhou, "Reliability analysis of multistate systems based on Bayesian networks," *Proc. of the 13th International Symposium and Workshop on Engineering of Computer Based Systems*, pp. 344–352, March 2006.