Analysis Methods of Multi-State Systems Partially Having Dependent Components Using Multiple-Valued Decision Diagrams

Shinobu Nagayama* Tsutomu Sasao† Jon T. Butler‡ Mitchell A. Thornton§ Theodore W. Manikas§

*Dept. of Computer and Network Eng., Hiroshima City University, Hiroshima, JAPAN
†Dept. of Computer Science, Meiji University, Kawasaki, JAPAN
‡Dept. of Electr. and Comp. Eng., Naval Postgraduate School, Monterey, CA USA
§Dept. of Computer Science and Engineering, Southern Methodist University, Dallas, Texas USA

***Abstract*—In a large system, such as a water, gas, or electrical distribution system, degraded performance due to failures of components can be modeled as a set of discrete states interconnected by edges with weights that represent conditional probabilities. To establish such a model, we compute the conditional probabilities with multi-valued decision diagrams (MDDs). Since a typical decision diagram is large, the computation time is also large. In this paper, we propose an edge-valued MDD (EVMDD) based method to avoid unnecessary path traversals. The proposed method is a hybrid method of a path traversal method and a bottom-up method that visits each node only once. By effectively combining both methods, we achieve a speed-up of the analysis by about 2.3 times for large systems compared to an existing method.**

***Keywords*-Multi-state systems with multi-state components; structure functions; system analysis based on decision diagrams; system analysis using conditional probabilities; EVMDDs.**

I. Introduction

Modern various systems, such as computer server systems, telecommunication systems, water, gas, and electrical power distribution systems, usually achieve tolerance to faults by redundancy. Even if a fault occurs, these systems still keep on working with an acceptable or degraded performance level. Thus, modeling such systems by two states (working and failure) is inadequate. They are often modeled by multi-state systems with multi-state components (or simply multi-state systems) in which levels of performance, reliability, safety, efficiency, power consumption, etc. are represented by more than two states [2], [3], [12], [14]–[16].

To design fault tolerant systems, intensive analysis of multi-state systems using various assessment measures for identifying critical components and system weaknesses is indispensable [12]. Among them, assessing the probability of each state of a multi-state system is essential to the design of a reliable fault tolerant system [14], [15]. Since this is very time-consuming, many methods to shorten analysis time have been proposed [1], [2], [4], [6], [10], [13]–[15]. Especially, methods based on decision diagrams have attracted much attention as fast analysis methods. Unlike conventional graphical models, such as Bayesian networks, that represent cause-and-effect relationships among components and a system, decision diagrams represent structure of a system as a function. Thus, information to be manipulated is reduced, and it results in faster analysis.

In these analysis methods of multi-state systems, it is assumed that states of components in the systems occur independently of other components. This reduces time complexity [2], [3] of the analysis. However, there exist systems, such as medical systems [7], in which states of components occur depending on other components. Such systems cannot be accurately analyzed by the methods assuming only independent events of components. We may overlook fatal flaws in systems by using such analysis methods. This should be avoided especially in safety-critical systems such as flight control systems and nuclear power plant monitoring systems. This paper shows fast and accurate analysis methods for multi-state systems having dependencies among components.

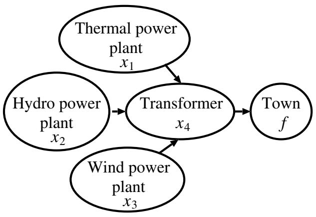
In [7], a multiple-valued decision diagram (MDD) based analysis method for such systems has been proposed. It can analyze systems quickly and accurately by traversing paths in an MDD even if systems have dependencies among components. However, systems do not always have dependencies among all components. There can be systems in which only a part of components has dependencies. Thus, in this paper, we propose a faster analysis method for those systems. The proposed method is based on an edge-valued MDD (EVMDD) [9], [10]. It avoids unnecessary path traversals of EVMDDs by applying a method intended for only independent events to components without dependencies, and analyzes systems more quickly and accurately.

This paper is organized as follows: Section II defines multi-state systems, EVMDDs, and some basic terminologies related to probability. Section III shows the analysis method based on path traversals in an EVMDD, and in Section IV, we propose a hybrid method to avoid unnecessary path traversals. Experimental results are shown in Section V.

II. Preliminaries

This section defines multi-state systems, structure functions, EVMDDs to represent structure functions, and basic terminologies related to probability.

0195-623X/14 $31.00 © 2014 IEEE
DOI 10.1109/ISMVL.2014.41



(a) Multi-state system.

x_1	x_2	x_3	x_4	f
0	0	0	0	0
0	0	0	1	0
0	0	0	2	0
0	0	1	0	0
0	0	1	1	1
		⋮		⋮
2	2	2	2	5

(b) Structure function.

Figure 1. Multi-state system for an electrical power distribution system and its structure function [11].

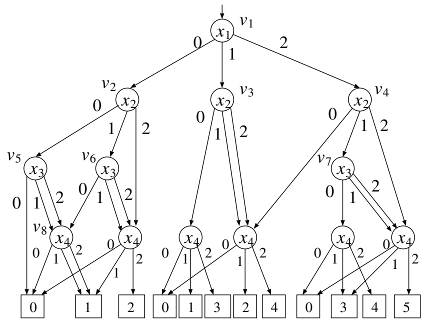


Figure 2. MDD for the structure function [11].

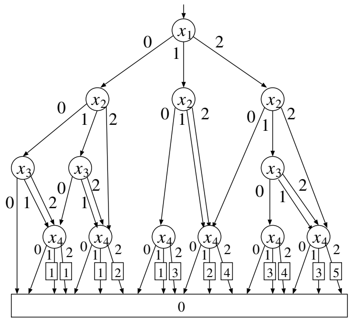


Figure 3. EVMDD for the structure function [11].

A. Multi-State Systems and Structure Functions

Definition 1: A **multi-state system** is a model of a system that represents, as states, a capability, such as performance, capacity, or reliability. There are usually more than two states, and so a multiple-valued analysis is required. When components in a system are modeled as well, it is called a **multi-state system with multi-state components**. In this paper, it is simply called a multi-state system.

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

Example 1: Fig. 1(a) shows a multi-state system for an electrical power distribution system. In this figure, the power plants x_1, x_2, x_3 and the transformer x_4 have three states which correspond to supply levels: 0 (breakdown), 1 (partially supply), and 2 (fully 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, we obtain the 6-valued structure function f shown in Fig. 1(b). Note that Fig. 1(b) shows a part of the $3^4 = 81$ entry table since it is too large to be included in its entirety. (End of Example)

B. Edge-Valued Multi-Valued Decision Diagrams

Definition 3: 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 [5]. It consists of non-terminal nodes representing sub-functions obtained from f by assigning values to certain variables. It also has terminal nodes representing sub-functions obtained from f by assigning values to all variables. These nodes are labeled by function values. Each non-terminal node has multiple outgoing edges that correspond to the values of a 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. The MDD is reduced by applying the following two reduction rules:

1) Share equivalent sub-graphs.
2) Delete non-terminal nodes whose outgoing edges all point to the same node.

When an MDD represents a function for which multi-valued variables have different domains, it is a heterogeneous MDD [8]. In the following, the term 'MDD' refers to a heterogeneous MDD.

Definition 4: An **edge-valued MDD (EVMDD)** [9] 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 the EVMDD, the following two reduction rules are applied:

1) Share equivalent sub-graphs.
2) Delete non-terminal nodes whose outgoing edges all point to the same node, and those edge weights are 0.

In the EVMDD, the function value is represented as a sum of weights for edges traversed from the root node to the terminal node.

Example 2: Fig. 2 and Fig. 3 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. (End of Example)

C. Conditional Probabilities

Definition 5: A **conditional probability**, denoted by $P(x_i = c_i | x_j = c_j)$, is the probability that an event $x_i = c_i$ occurs given that an event $x_j = c_j$ has occurred. The following holds:

$$\sum_{c_i=0}^{r_i-1} P(x_i = c_i | x_j = c_j) = 1. \tag{1}$$

Definition 6: A **joint probability**, denoted by $P(x_i = c_i, x_j = c_j)$, is the probability that both of an event $x_i = c_i$ and an event $x_j = c_j$ occur.

The conditional probability and the joint probability can be extended to more than two events. The following relations hold between the conditional probability and the joint probability:

$$P(x_i = c_i, x_j = c_j) = P(x_i = c_i | x_j = c_j) \cdot P(x_j = c_j). \tag{2}$$

$$\begin{aligned} P(x_i = c_i) &= \sum_{c_j=0}^{r_j-1} P(x_i = c_i, x_j = c_j), \\ &= \sum_{c_j=0}^{r_j-1} P(x_i = c_i | x_j = c_j) \cdot P(x_j = c_j) \end{aligned} \tag{3}$$

Definition 7: If an event $x_i = c_i$ does not directly depend on an event $x_j = c_j$ (i.e., the events are **conditionally independent**), then the following relation holds:

$$\begin{aligned} & P(x_i = c_i, x_j = c_j | x_k = c_k) \\ = \; & P(x_i = c_i | x_k = c_k) \cdot P(x_j = c_j | x_k = c_k). \end{aligned}$$

Also,

$$P(x_i = c_i | x_j = c_j, x_k = c_k) = P(x_i = c_i | x_k = c_k).$$

In general, thus, the joint probability is computed by:

$$\begin{aligned} P(x_1 = c_1, x_2 = c_2, \ \ldots \ , x_n = c_n) \\ = P(x_1 = c_1 | \Pi(x_1)) \cdot P(x_2 = c_2 | \Pi(x_2)) \cdot \\ \ldots \cdot P(x_n = c_n | \Pi(x_n)), \end{aligned} \tag{4}$$

where $\Pi(x_i)$ is a set of variables on which x_i depends and values assigned to them.

III. Path Traversal Based Analysis Methods

Definition 8: The probability that a structure function f has the value s is denoted by $P_s(f = s)$, where $s \in \{0, 1, \ldots, m-1\}$.

An analysis of multi-state systems is to solve the following problem:

Problem 1: Given a structure function f of a multi-state system and the conditional probability of each state of each component, compute the probability of each state of the multi-state system $P_s(f = s)$.

From the definition of the joint probability, the probability of each system state $P_s(f = s)$ can be computed as follows:

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

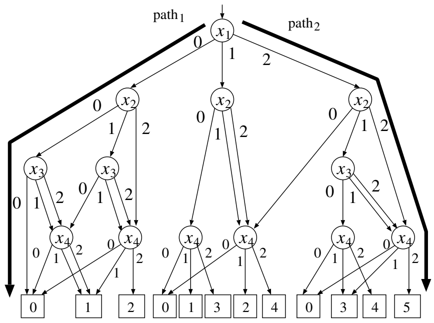


Figure 4. Path traversals in the MDD.

where R^n is a set of value assignments to all the variables x_i, and $P(f = s, x_1, x_2, \ldots, , x_n)$ is a joint probability of $f = s$ and $(x_1, x_2, \ldots, x_n) = \vec{c} = (c_1, c_2, \ \ldots, c_n)$. From (2), we have

$$\begin{aligned} & P(f = s, x_1 = c_1, x_2 = c_2, \ldots, x_n = c_n) = \\ & \quad P(f = s | x_1 = c_1, x_2 = c_2, \ldots, x_n = c_n) \cdot \\ & \quad\quad P(x_1 = c_1, x_2 = c_2, \ldots, x_n = c_n), \end{aligned}$$

where

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

because f is a function of $x_1, x_2, \ldots, x_n$. Therefore, (5) is obtained by a sum of joint probabilities that an input vector $\vec{c}$ satisfies $f = s$. Since each joint probability can be computed by (4), the time complexity of this straightforward method is $O(r^n)$ where all x_i are r-valued variables.

A. Method Based on Path Traversals in MDD

To solve Problem 1 more efficiently, a method that analyzes systems by traversing paths in an MDD has been proposed [7]. Since the number of paths in an MDD becomes smaller than r^n due to the reduction rule for deleting non-terminal nodes, this analysis method is faster.

Example 3: In this example, assume that x_4, x_3, and x_2 depend on x_3, x_2, and x_1, respectively. Then, consider path$_1$ in Fig. 4. Since this path does not visit a node for x_4, from the reduction rule for deleting the node, we compute the following using conditional probabilities assigned to each edge and (1):

$$\begin{aligned} & \sum_{c_4=0}^{2} P(x_4 = c_4 | x_3 = 0) \cdot \\ & \quad P(x_1 = 0) \cdot P(x_2 = 0 | x_1 = 0) \cdot P(x_3 = 0 | x_2 = 0), \\ = \; & 1 \cdot P(x_1 = 0) \cdot P(x_2 = 0 | x_1 = 0) \cdot P(x_3 = 0 | x_2 = 0), \\ = \; & P(x_1 = 0, x_2 = 0, x_3 = 0). \end{aligned}$$

Similarly, since path$_2$ does not visit a node for x_3, we

Algorithm 1: (Path traversal method of EVMDDs)

```
 1:path_EVMDD (node v, vector c⃗, edge value e, probability p) {
 2:   if (v is the terminal node) {
 3:      prob_sys_state[e] += p;
 4:      return;
 5:   }
 6:   r_i = domain size of variable x_i for v;
 7:   for (j = 0; j < r_i; j++) {
 8:      c_i = j;
 9:      p' = p * cond_prob(x_i, c⃗, Π(x_i));
10:      path_EVMDD(j-th edge of v, c⃗, e + j-th edge value of v, p');
11:   }
12:   c_i = undefined;
13:}
```

compute the following using (3):

$$\begin{aligned}
&\sum_{c_3=0}^{2} P(x_3=c_3|x_2=2)\cdot P(x_4=2|x_3=c_3)\cdot \\
&\qquad P(x_1=2)\cdot P(x_2=2|x_1=2), \\
&= \sum_{c_3=0}^{2} P(x_1=2,x_2=2,x_3=c_3,x_4=2), \\
&= P(x_1=2,x_2=2,x_4=2).
\end{aligned}$$

In this way, we can analyze multi-state systems having any dependencies among components by computing a probability on each path, and accumulating it at each terminal node. (End of Example)

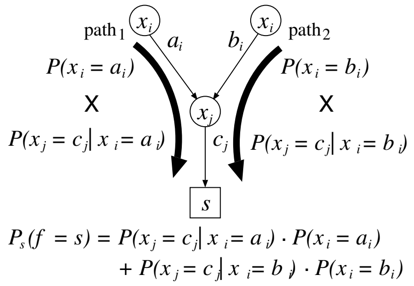
B. Method Based on Path Traversals in EVMDD

In this subsection, we extend the MDD-based method shown in the previous subsection to the EVMDD-based one. The main difference is only the handling of edge values. While the MDD-based method accumulates probabilities on paths at each terminal node, the EVMDD-based method accumulates probabilities on paths in each sum of edge values. Algorithm 1 shows a pseudo-code for the analysis method based on path traversals in an EVMDD.

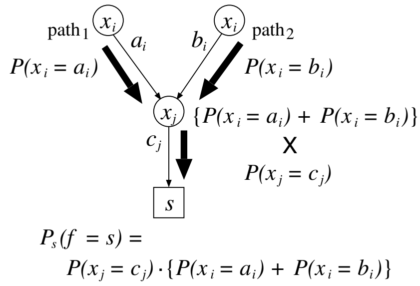
This algorithm requires the root node of an EVMDD, the empty vector in which all elements are undefined, the edge value for the root node, and the probability for the root node 1.0 as input parameters, and then it traverses all paths from the root node to the terminal node recursively. In the 3rd line, the array prob_sys_state stores a probability for each system state. In the 9th line, the procedure **cond_prob**(x_i, $\vec{c}$, $\Pi(x_i)$) computes a conditional probability $P(x_i = c_i|\Pi(x_i))$ along with conditional probabilities for skipped parent nodes, as shown in Example 3. Note that a variable x_i depending on other variables $\Pi(x_i)$ has to be below those variables $\Pi(x_i)$ in the variable order of EVMDD to compute conditional probabilities.

IV. Hybrid Analysis Method for Systems Partially Having Dependent Components

Although Algorithm 1 can analyze multi-state systems having any dependencies among components, systems do not always have dependencies among all components. There can be systems in which only some of the components have dependencies. For analysis of such systems, Algorithm 1, which is based on path traversals in an EVMDD is not very efficient. This is because the path traversal based methods can visit a node many times that



(a) x_j depends on x_i.



(b) x_j is independent.

Figure 5. Computation of conditional probabilities in EVMDDs.

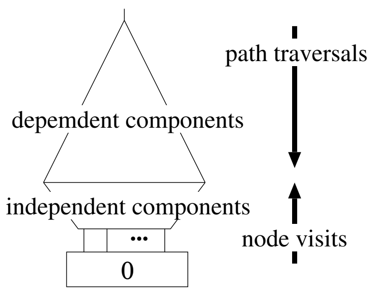


Figure 6. Partition of EVMDD.

is shared due to the reduction rule for sharing equivalent sub-graphs. As shown in Fig. 5(a), when x_j depends on x_i, a conditional probability has to be computed in each path since

$$\begin{aligned}
&P(x_j=c_j|x_i=a_i)\cdot P(x_i=a_i)+P(x_j=c_j|x_i=b_i)\cdot P(x_i=b_i) \\
&\neq \quad P(x_j=c_j|x_i=a_i)\cdot\{P(x_i=a_i)+P(x_i=b_i)\}.
\end{aligned}$$

On the other hand, when x_j is independent of others as shown in Fig. 5(b), probabilities obtained from parent nodes can be tallied at each node since

$$\begin{aligned}
&P(x_j=c_j)\cdot P(x_i=a_i)+P(x_j=c_j)\cdot P(x_i=b_i) \\
&= \quad P(x_j=c_j)\cdot\{P(x_i=a_i)+P(x_i=b_i)\}.
\end{aligned}$$

Thus, we can apply a method that visits each node only once to components without dependencies to avoid unnecessary path traversals of EVMDDs.

Suppose that an EVMDD can be partitioned into two parts: the upper and the lower parts, as shown in Fig. 6, where the upper part has components that depend on

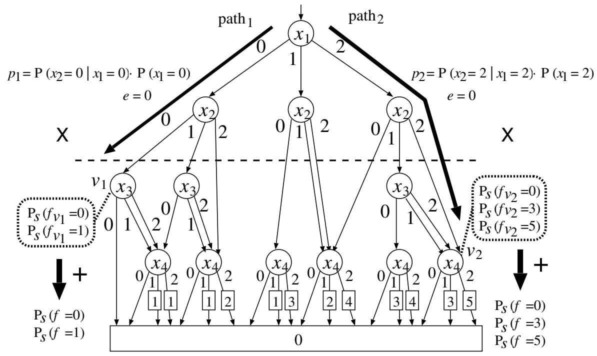


Figure 7. Analysis using the hybrid method.

others, and the lower has only independent components*. Then, we apply the path traversal based method to the upper part, and apply the node visits based method [10] to the lower part in a bottom-up manner. The method [10] visits each node only once in a bottom-up manner. And at each node, it computes probabilities for its sub-function by merging probabilities obtained at child nodes.

We begin by applying the node visit based method to the lower part. Although in [10], the method is applied to all components from the terminal node to the root node, in the hybrid method, it is applied only to independent components. Then, we apply the path traversal based method to the upper part. When a path traversal reaches a node v in the lower part, the probability p on the path is multiplied by probabilities $P_s(f_v = s)$ for the sub-function of v that are obtained by the node visit based method. The products $p \cdot P_s(f_v = s)$ are accumulated as probabilities $P_s(f = s + e)$ of system states $s + e$, where e is a sum of edge values the path traversed.

Example 4: In this example, assume that x_2 depends on x_1, and there are no other dependencies. Then, as shown in Fig. 7, nodes for x_1 and x_2 are in the upper part, and nodes for x_3 and x_4 are in the lower part. First, probabilities at each node in the lower part are computed in a bottom-up manner [10]. Next, probabilities on paths from the root node to nodes in the lower part are computed. Consider path$_1$ in Fig. 7. This path reaches node v_1, and thus the probability $p_1 = P(x_2 = 0|x_1 = 0) \cdot P(x_1 = 0)$ on the path is multiplied by probabilities $P_s(f_{v_1} = 0)$ and $P_s(f_{v_1} = 1)$ at v_1, respectively. The products $p_1 \cdot P_s(f_{v_1} = 0)$ and $p_1 \cdot P_s(f_{v_1} = 1)$ are accumulated as probabilities $P_s(f = 0)$ and $P_s(f = 1)$, respectively, since the sum of edge values e for the path is 0. Similarly, since path$_2$ reaches node v_2, the products $p_2 \cdot P_s(f_{v_2} = 0)$, $p_2 \cdot P_s(f_{v_2} = 3)$, and $p_2 \cdot P_s(f_{v_1} = 5)$ are accumulated as probabilities $P_s(f = 0)$, $P_s(f = 3)$, and $P_s(f = 5)$, respectively.

In this way, we analyze systems having partially dependent components efficiently by combining the path traversal and node visit based methods. (End of Example)

*If the lower part has dependent components, the variable order is changed so that the lower part has only independent components.

Computation on each path in the path traversal based method is simpler than computation at each node in the node visit based method. Thus, the path traversal based method is faster when the number of paths is not much larger than that of nodes (i.e., when the number of shared nodes is small). As the shape of the EVMDD in Fig. 6 shows, nodes in the upper part are hardly shared while nodes in the lower part tend to be shared. Thus, by applying the path traversal and the node visit based methods to the upper and the lower parts, respectively, we can analyze systems efficiently. In the experiment in Section V, we partitioned an EVMDD with n components into the upper part with $\lceil \frac{2}{3}n \rceil$ components and the lower part with $\lfloor \frac{n}{3} \rfloor$ components because it produces the best performance for a large system.

V. Experimental Results

To show the efficiency of the proposed hybrid method, we compare it with various existing methods using the same structure functions as [10]. This is because unfortunately, benchmark structure functions of large enough multi-state systems are unavailable. In this experiment, we analyze the multi-state systems given by those structure functions, assuming that only a component x_3 depends on another component x_2. The methods are implemented on our private EVMDD package, and run on the following computer environment: CPU: Intel Core2 Quad Q6600 2.4GHz, memory: 4GB, OS: CentOS 5.7, and C-compiler: gcc -O2 (version 4.1.2). Table I shows the experimental results for randomly generated m-state systems with n 3-state components.

From this table, we can see that the methods based on path traversals in MDDs and EVMDDs are much faster than the naive method that is used in Bayesian networks [16]. This is because the number of paths in MDDs and EVMDDs is much smaller than 3^n. The method based on path traversals in EVMDDs is slightly slower than that in MDDs because of the overhead to compute edge values.

For large systems, the hybrid method using both path traversal and node visit based methods is faster than the path traversal based methods. Especially, as a difference between the number of paths and the number of nodes in an EVMDD becomes larger, the hybrid method becomes faster. In this experiment, when the number of nodes is smaller than $\frac{1}{3}$ of the number of paths, the hybrid method is faster than the path traversal based methods. On the other hand, in ordinary MDDs, the number of nodes is not much smaller than the number of paths, since the number of shared nodes is small. Thus, in this experiment, we could not achieve a speed-up by applying the hybrid method to MDDs.†

From the above observations, we can conclude that the efficiency of the hybrid method is due in large part to the compactness of EVMDDs, and it is the method that can take advantage of the fact that EVMDDs are compact.

†Results about those are omitted.

Table I
COMPUTATION TIMES FOR ANALYSIS OF m-STATE SYSTEMS WITH n 3-STATE COMPONENTS.

n	m	Number of nodes		Number of paths			Computation time (μsec.)				
		MDD	EVMDD	3^n	MDD	EVMDD	Naive (5)	Path traversals methods		Hybrid	Ratio
								MDD [7]	EVMDD		
5	3	12	10	243	19	19	10.08	1.36	1.47	1.77	131%
5	10	36	18	243	53	53	10.07	3.39	3.57	4.76	140%
10	3	17	15	59,049	29	29	4,030.00	1.98	2.01	2.74	139%
10	10	77	57	59,049	135	135	4,019.00	8.83	9.12	12.14	137%
10	100	599	265	59,049	999	999	4,011.00	63.28	66.35	69.60	110%
10	1,000	4,201	907	59,049	6,403	6,403	4,031.00	419.51	437.76	360.14	86%
15	3	32	30	14,348,907	59	59	1,356,000.00	3.77	3.91	5.95	158%
15	10	120	105	14,348,907	221	221	1,358,000.00	14.04	14.84	21.14	151%
15	100	1,098	708	14,348,907	1,997	1,997	1,359,000.00	124.90	132.65	148.64	119%
15	1,000	9,010	3,363	14,348,907	16,021	16,021	1,360,000.00	1,014.94	1,086.62	930.28	92%
15	10,000	70,140	11,474	14,348,907	120,281	120,281	1,360,000.00	7,781.00	8,230.00	5,408.00	70%
15	100,000	495,224	62,759	14,348,907	790,449	790,449	1,359,000.00	55,786.00	54,801.00	24,238.00	43%

n: Number of 3-state components. m: Number of states for systems. Ratio: Hybrid / MDD [7] $\times$ 100 (%)
Naive (5): The method that directly computes (5).
Very short computation times are average times obtained by running the same computation $1,000,000$ times, and dividing its total time by $1,000,000$.

VI. Conclusion and Comments

This paper proposes an efficient analysis method of multi-state systems in which states of some components occur depending on states of other components. The proposed method is a hybrid method using both path traversals and node visits in an EVMDD. By using the node visit based method, we can avoid unnecessary path traversals and achieve fast analysis. Experimental results show that we can analyze large systems quickly and accurately using the proposed hybrid method. For a large system, we achieve a speed-up of 2.3 times, compared to the existing path traversal based method. For future work, we will study a heuristic method to find a better partition of an EVMDD in Fig. 6 to improve performance of the hybrid method.

Acknowledgments

This research is partly supported by the JSPS Grant-in-Aid for Scientific Research (C), (No. 25330071), 2014.

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," *the 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] 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.

[6] T. W. Manikas, M. A. Thornton, and D. Y. Feinstein, "Using multiple-valued logic decision diagrams to model system threat probabilities," *41th International Symposium on Multiple-Valued Logic*, pp. 263–267, May 2011.

[7] 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.

[8] S. Nagayama and T. Sasao, "On the optimization of heterogeneous MDDs," *IEEE Trans. on CAD*, Vol. 24, No. 11, pp. 1645–1659, Nov. 2005.

[9] 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.

[10] 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.

[11] S. Nagayama, T. Sasao, and J. T. Butler, "Minimization of the number of edges in an EVMDD by variable grouping for fast analysis of multi-state systems," *43rd International Symposium on Multiple-Valued Logic*, pp.284-289, May, 2013.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] Z. Zhou, G. Jin, D. Dong, and J. Zhou, "Reliability analysis of multistate systems based on Bayesian networks," *the 13th International Symposium and Workshop on Engineering of Computer Based Systems*, pp. 344–352, March 2006.