OBDD-Based Evaluation of Reliability and Importance Measures for Multistate Systems Subject to Imperfect Fault Coverage

Yung-Ruei Chang, *Member*, *IEEE*, Suprasad V. Amari, *Senior Member*, *IEEE*, and Sy-Yen Kuo, *Fellow*, *IEEE*

Abstract—Algorithms for evaluating the reliability of a complex system such as a multistate fault-tolerant computer system have become more important recently. They are designed to obtain the complete results quickly and accurately even when there exist a number of dependencies such as shared loads (reconfiguration), degradation, and common-cause failures. This paper presents an efficient method based on Ordered Binary Decision Diagram (OBDD) for evaluating the multistate system reliability and the Griffith's importance measures which can be regarded as the importance of a system-component state of a multistate system subject to imperfect fault-coverage with various performance requirements. This method combined with the conditional probability methods can handle the dependencies among the combinatorial performance requirements of system modules and find solutions for multistate imperfect coverage model. The main advantage of the method is that its time complexity is equivalent to that of the methods for perfect coverage model and it is very helpful for the optimal design of a multistate fault-tolerant system.

Index Terms— Reliability, multistate system, OBDD, fault-coverage, importance measure.

1 Introduction

The multistate system theory has been investigated since 1975 [1]. For example, a power plant which has states 0, 1, 2, 3, and 4 that correspond to generating electricity of 0, 25, 50, 75, 100 percent of its full capacity is an example of a multistate system that has ordered multiple states [2]. A nuclear reactor system [3] or a pumping system [4] that performs differently according to the many different combinations of states of its subsystems or submodules are example multistate systems with unordered multiple states. Many researchers have analyzed the multistate system reliability [5], [6], [7]. Most of them extend the concepts and conclusions for the two-state systems to the multistate systems. To describe the dynamic characteristics of the component state transition, Stochastic process (Markov process) techniques are combined with the multistate system theory to analyze the dynamic multistate system reliability. The multistate reliability theory can handle situations in which the system and its components have a range of performance levels, e.g., from perfect operation to complete failure. Because performance degradation is very common in industrial products, it is important to develop the multistate system reliability theory.

When a multistate system (MSS) is considered, it is important to estimate the impact of each element on the system output/performance. The general definition of the MSS reliability [6] is:

$$R_{MSS}(t, L) = \Pr\{F(t) \geq L\}, \quad (1)$$

where L is the required performance level for MSS, $F(t)$ is the MSS output/performance at time t, which is a random variable. That is, $R_{MSS}(t, L)$ means the probability that the system output/performance is above the required performance level over the time interval $(0, t)$. For a multistate system that has a finite number of states, there can be H different levels of output/performance at time t. Therefore, $F(t)$ belongs to the finite vector set $\mathbf{F}$ which is composed of H different levels of output/performance, F_h:

$$F(t) \in \mathbf{F} = \{F_h, 1 \leq h \leq H\} \quad (2)$$

and the system output/performance distribution can be defined by the following finite vector set:

$$\mathbf{q} = \{q_h(t)\} = \{\Pr\{F(t) = F_h\}\}, \quad (1 \leq h \leq H). \quad (3)$$

Therefore, the nonrepairable MSS reliability, which is also equivalent to availability, is the probability that the system remains in the states with $F_h \geq L$ during $(0, t)$:

$$R_{MSS}(t, L) = \sum_{F_h \geq L} q_h(t). \quad (4)$$

Similarly, (4) could be used for the definition of MSS availability if the multistate system is repairable, where $q_h(t)$ is the availability of the system running at the performance F_h.

In addition, systems that are used in life-critical applications such as flight control, nuclear power plant monitoring,

- *Y.-R. Chang is with Institute of Nuclear Energy Research, Atomic Energy Council, Taoyuan 32546, Taiwan. E-mail: raymond@iner.gov.tw.*
- *S.V. Amari is with Relex Software Corporation, 540 Pellis Road, Greensburg, PA 15601. E-mail: suprasad.amari@relexsoftware.com.*
- *S.-Y. Kuo is with the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan. E-mail: sykuo@cc.ee.ntu.edu.tw.*

Manuscript received 2 Feb. 2004; revised 18 July 2005; accepted 6 Oct. 2005; published online 3 Nov. 2005.
For information on obtaining reprints of this article, please send e-mail to: tdsc@computer.org, and reference IEEECS Log Number TDSC-0025-0204.

space missions, etc., are designed with sufficient redundancy to be tolerant of errors. However, if the system cannot adequately detect, locate, and recover from faults and errors in the system, then system failure can still occur even when there exists adequate redundancy [8]. An accurate analysis must account for not only the complex system structure, but also the system fault and error recovery behavior. Therefore, the fault-coverage problem of a system should be considered. This helps in determining the optimal level of redundancy [9].

Most of the published works use Markov models (nonhomogenous Markov or semi-Markov model) to solve multistate problems. However, it is difficult to find the correct model of a system since there will be a total of $N = (m+1)^n$ states if there are n modules in the system and each module has $(m+1)$ states including the imperfect coverage state. The computational time to solve the Markov model is proportional to $N^3 = [(m+1)^n]^3$. Hence, the computational complexity of the problem is $O(m^{3n})$ [10], [11].

The recent literature [12], [13], [14], [15], [16], [17], [18], [19] showed that, in many cases, Ordered Binary Decision Diagram (OBDD)-based algorithms are more efficient in reliability evaluation compared to other methods such as the Inclusion-Exclusion (I-E) method and the sum of disjoint products (SDP) method. This paper, which is modified from [14], provides an approach for modelling a multistate system and proposes an efficient method integrated with the OBDD method and the conditional probability concepts to evaluate the reliability of a multistate system with imperfect coverage. This efficient integration of the OBDD and the modularization methods simplifies the problem further. This method could also be integrated with those methodologies that use the OBDD method for reliability analysis [13], [15], [16], [17], [18].

Furthermore, the importance measure, also called the sensitivity analysis, is used to measure the effect of the individual component reliabilities on the system reliability. Identifying the weaknesses of the system and how failure of each individual component affects the proper functioning of the system is an important topic for optimal design issue so that efforts can be spent properly to improve the system reliability [20]. Several literatures have introduced the importance measures in a two-state system. Birnbaum-importance measure indicates the contribution of component reliability to the system reliability [20], [21]. Structural-importance measure indicates the topographic importance of a position in the system [22], [23]. Criticality-importance measure corresponds to the conditional probability of a component failure, given that the system has failed [20], [21]. Joint-importance measure indicates how components in a system interact and contribute to the system-reliability [24], [25].

Very few publications discuss how the particular state of a component contributes to a multistate system, and how the presence of a component and a particular state of a component affects the contributions of other components in the system. Most importance concepts have been built on how the state change of a component affects the system [26], [27], [28], [29], [30], [31], [32]. In this paper, we also propose an OBDD-based method to compute the Griffith importance vector [31], which can be interpreted as the range of a multistate system performance when a module moves from one state to another state and can be regarded as the importance of the state of the system-module. Moreover, with a little modification, we can extend our method to evaluate the Griffith's importance vector under the imperfect fault-coverage model.

Section 2 introduces the fundamentals of OBDD and coverage model. Section 3 illustrates a multistate imperfect coverage model and an OBDD-based approach to evaluating the reliability and availability of a multistate system with imperfect coverage. In Section 4, an OBDD-based approach is introduced to compute the Griffith's importance measure subject to either perfect or imperfect fault-coverage. Section 5 gives some examples. The last section gives the conclusions and future works.

2 Preliminaries

2.1 Ordered Binary Decision Diagram (OBDD)

This section introduces the representation and manipulation of Boolean functions based on OBDD. OBDD [12] is based on a decomposition of Boolean function called the *Shannon expansion*. A function f can be decomposed in terms of a variable x as:

$$f = x \cdot f_{x=1} + \bar{x} \cdot f_{x=0}. \tag{5}$$

A node and its descendants in an OBDD represent a Boolean function f, where, for node label x, one outgoing edge is directed to the subgraph representing $f_{x=1}$ and the other to $f_{x=0}$. Shannon decomposition is the basis for using OBDD. In order to express Shannon decomposition concisely, the if-then-else (*ite*) format [33], [34] is defined as:

$$f = ite(x, f_{x=1}, f_{x=0}). \tag{6}$$

The manipulation of OBDD to represent logical operations is simple. In practice, the OBDD is generated by using logical operations on variables. Let Boolean expressions f and g be:

$$\begin{aligned} f &= ite(x, f_{x=1}, f_{x=0}) = ite(x, F_1, F_0) \\ g &= ite(y, g_{y=1}, g_{y=0}) = ite(y, G_1, G_0). \end{aligned} \tag{7}$$

A logic operation between f and g can be represented by OBDD manipulations as:

$$\begin{aligned} &ite(x, F_1, F_0) \diamondsuit ite(y, G_1, G_0) = \\ &\begin{cases} ite(x, F_1 \diamondsuit G_1, F_0 \diamondsuit G_0) & \text{ordering}(x) = \text{ordering}(y) \\ ite(x, F_1 \diamondsuit g, F_0 \diamondsuit g) & \text{ordering}(x) < \text{ordering}(y) \\ ite(y, f \diamondsuit G_1, f \diamondsuit G_0) & \text{ordering}(x) > \text{ordering}(y), \end{cases} \end{aligned} \tag{8}$$

where $\diamondsuit$ represents a logic operation such as AND and OR. Fig. 1 illustrates the construction and manipulation steps of a Boolean function. For more details on using the operations of OBDD, please refer to [12].

2.2 Coverage Model

Fig. 2a shows the general structure of a fault-coverage model representing a recovery process [35], [36] initiated when a fault occurs. The entry point to the model signifies

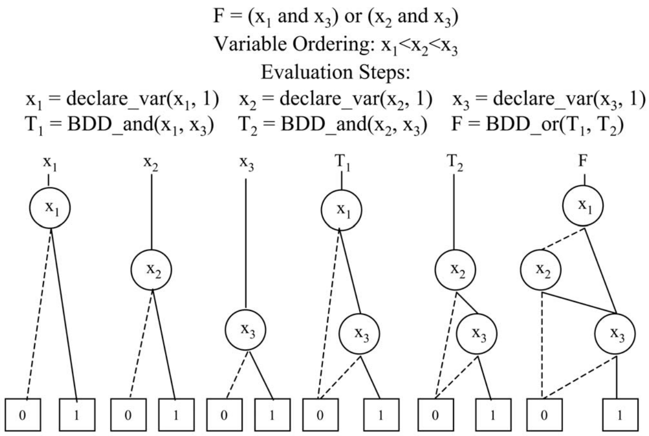


Fig. 1. The OBDD generated from a Boolean equation.

the occurrence of a fault, and the three exits (R, S, C) signify the three possible outcomes:

- If the offending fault is transient and can be handled without discarding any components, then the transient restoration exit (R) is taken.
- If the fault is determined to be permanent, and the offending component is discarded, then the permanent fault-coverage exit (C) is taken.
- If the fault by itself causes a system to fail, then the single-point failure exit (S) is taken.

The exit probabilities r_0, c_0, s_0 are required for the analysis of system reliability. The exits are a partitioning of the event space; thus, the three exit probabilities sum to one, i.e., $(c_0 + s_0) = (1 - r_0)$. The values of r_0, c_0, s_0 can be determined by an appropriate fault-coverage model [36]; for more details, see [8], [37].

For the fault-coverage model, each component is always in one of three states: $x[i]$, $y[i]$, $z[i]$, which represent, in Fig. 2b, the states of *component not failed*, *component failed and covered*, and *component failed and uncovered*, respectively. To determine the system reliability (unreliability), it is required to have $a[i], b[i], c[i]$ that represent the probabilities of component i associated with the three exits of the fault-coverage model, respectively. Fig. 2b shows the event space (and corresponding probability) representation of a component. Therefore,

$$\begin{aligned} a[i] &= \exp[-(1 - r_{i0}) \cdot \lambda_{i0} \cdot t] \\ b[i] &= \frac{c_{i0}}{c_{i0} + s_{i0}} \cdot [1 - \exp[-(1 - r_{i0}) \cdot \lambda_{i0} \cdot t]] \\ c[i] &= \frac{s_{i0}}{c_{i0} + s_{i0}} \cdot [1 - \exp[-(1 - r_{i0}) \cdot \lambda_{i0} \cdot t]], \end{aligned} \tag{9}$$

where (r_{i0}, c_{i0}, s_{i0}) are the probabilities of taking (transient restoration, permanent coverage, and single-point failure) exit, respectively, in the coverage model and λ_{i0} is the fault occurrence rate of component i. It should be noted that the effective failure rate λ_i and the effective coverage factor c_i of component i are

$$\begin{aligned} \lambda_i &\equiv (c_{i0} + s_0) \cdot \lambda_{i0} = (1 - r_{i0}) \cdot \lambda_{i0} \\ c_i &\equiv c_{i0}/(c_{i0} + s_{i0}). \end{aligned} \tag{10}$$

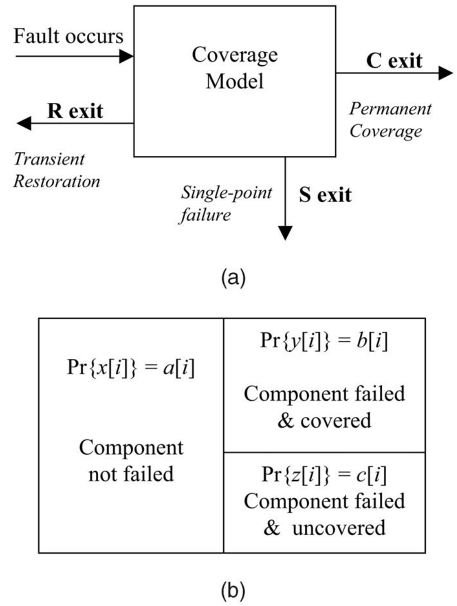


Fig. 2. (a) General structure of a fault-coverage model. (b) The event and probability space of component i.

When a system consists of multiple components and each component is subject to imperfect fault-coverage, the reliability evaluation of such a system becomes more complicated. Amari et al. [35] proposed an efficient algorithm, the SEA, to calculate the reliability of a system that is composed of multiple components under the imperfect coverage model (IPCM). The basic idea is shown in (11) and could be easily proven [35] by using conditional probabilities or the total probability theorem.

$$\begin{aligned} &\text{System Unreliability } (U_s(t)) \\ &= \Pr\{\text{at least one uncovered failure}\} \\ &\quad \times \Pr\{\text{system failure} \mid \text{at least one uncovered failure}\} \\ &+ \Pr\{\text{no uncovered failure}\} \\ &\quad \times \Pr\{\text{system failure} \mid \text{no uncovered failure}\}. \end{aligned} \tag{11}$$

Let $\Pr\{\text{no uncovered failure}\} = \Pi_{i \in S}(a[i] + b[i]) = P_u(t)$, then $\Pr\{\text{at least one uncovered failure}\} = 1 - P_u(t)$. Also, let $\Pr\{\text{system failure} \mid \text{no uncovered failure}\} = U_s^c(t)$. Since

$$\Pr\{\text{system failure} \mid \text{at least one uncovered failure}\}$$

is always equal to 1, we have

$$\begin{aligned} U_s(t) &= [1 - P_u(t)] + P_u(t) \cdot U_s^c(t) = 1 - P_u(t) \cdot R_s^c(t) \\ R_s(t) &= 1 - U_s(t) = P_u(t) \cdot R_s^c(t), \end{aligned} \tag{12}$$

where $R_s(t)$ is the system reliability and $R_s^c(t)$ is $\Pr\{\text{system success} \mid \text{no uncovered failure}\}$. $R_s^c(t)$ can be derived by using the conditional reliability instead of the reliability of each component during the evaluation of the conditional system reliability. The correctness and the proof are given in [35]. Therefore, the time complexity of (12) will be equivalent to that of the method under perfect coverage model (PCM).

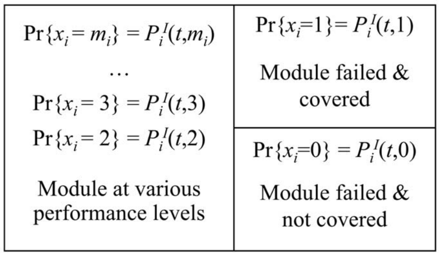


Fig. 3. The probability space of multistate module i.

3 Multistate Coverage Model

3.1 Multistate Systems with Imperfect Coverage

In this section, we will extend the three-state model as depicted in Section 2 into a multistate imperfect fault-coverage model. Assume there are n modules in a system, module i has m_i states $(i = 1, \ldots, n)$, and a system has H different levels of output/performance. Depending upon the performance/capacity, we can arrange the states such that state m_i is a perfect state and state 1 is a failed state (the performance level decreases from state m_i to state 1). The ordering is not a constraint, but it helps our algorithm to include the existing algorithms that are for a multistate coherent system under the perfect coverage model (PCM). However, for the imperfect coverage model (IPCM), each module will have an extra state, i.e., the total number of states in module i becomes $m_i + 1$. Here, state 0 is the state corresponding to the uncovered failure of module i. Fig. 3 shows the event and probability space of the multistate module i.

Assumption. The uncovered failure of any module within a system causes immediate uncovered failure of the system.

State Representation

state H: perfect state of a system (highest performance level of a system)
state h: state of a system at performance level h
state m_i: perfect state of module i (highest performance level of module i)
state j: state of a module at performance level j
state 1: state of a system/module at zero performance level (state of system/module failed and covered)
state 0: state of system/module failed and uncovered

Notation

x_i, x_i^I indicator variable of state of module i under [PCM, IPCM]; $x_i = 1$ means modules i under PCM is in state l.
$x_{i:j}, x_{i:j}^I$ represents that module i under [PCM, IPCM] is at performance level j or above, i.e., $x_{i:j}, x_{i:j}^I$ are equivalent to $x_i \geq j, x_i^I \geq j$
$P_i(t,j)$ Pr{module i under PCM is in state j at time t}
$P_i^I(t,j)$ Pr{module i under IPCM is in state j at time t}
$P_i^c(t,j)$ Pr{module i under IPCM is in state j at time t | no uncovered failure in module i}
$R_i(t,j)$ $\Pr\{\text{module } i \text{ under PCM is in state} \geq j \text{ at time } t\} = \sum_{k=j}^{m_i} P_i(t,k) = \Pr\{x_{i:j}\}$
$R_i^I(t,j)$ $\Pr\{\text{module } i \text{ under IPCM is in state} \geq j \text{ at time } t\} = \sum_{k=j}^{m_i} P_i^I(t,k) = \Pr\{x_{i:j}^I\}$
$R_i^c(t,j)$ $\Pr\{\text{module } i \text{ under IPCM is in state } \geq j \text{ at time } t \mid \text{no uncovered failure in module } i\}$
$P_s(t,h)$ $\Pr\{\text{system under PCM is in state } h \text{ at time } t\}$
$P_s^I(t,h)$ $\Pr\{\text{system under IPCM is in state } h \text{ at time } t\}$
$P_s^c(t,h)$ $\Pr\{\text{system under IPCM is in state } h \text{ at time } t \mid \text{no uncovered failure in the system}\}$
$R_s(t,h)$ $\Pr\{\text{system under PCM is in state } \geq h \text{ at time } t\} = \sum_{k=h}^{H} P_s(t,k)$
$R_s^I(t,h)$ $\Pr\{\text{system under IPCM is in state} \geq h \text{ at time } t\} = \sum_{k=h}^{H} P_s^I(t,k)$
$R_s^c(t,h)$ $\Pr\{\text{system under IPCM is in state} \geq h \text{ at time } t \mid \text{no uncovered failure in the system}\}$

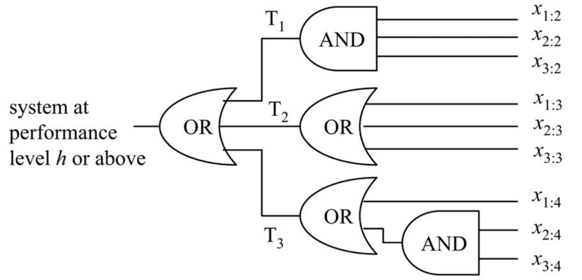


Fig. 4. The combination of performance requirements of a system being operational at performance level h or above.

Fig. 4 shows a combination of performance requirements of a multistate system for being operational at performance level h or above. It includes three subrequirements T_1, T_2, and T_3. $x_{i:j}$ is the basic event of subrequirements and represents the minimum performance requirement of module i. That is, $x_{i:j}$ means module i needs to be operational at performance level j or above. For example, in Fig. 4, the system will be at performance level h or above if every module i $(i = 1, 2, 3)$ is operational at level 2 or above (T_1), or if any module i $(i = 1, 2, 3)$ is operational at level 3 or above (T_2), or if module 1 is operational at level 4 or above, or both module 2 and module 3 are operational at level 4 or above (T_3).

There exist several methods to solve the problem of reliability evaluation of a multistate system that includes various combinations of subrequirements. However, when there exists imperfect coverage, the entire problem should be solved using Markov chains even when the system requirements can be represented as a combination of modular subrequirements. For large complex systems, the Markov is not suitable for the well-known exponential state explosion problem. Therefore, there is a reasonable method to solve MSS reliability under IPCM. In the next section, we will propose an efficient method for evaluating the reliability of a multistate system subject to imperfect fault-coverage. The computational complexity of the method is equivalent to that of the method under PCM.

3.2 Reliability/Availability Evaluation of a Multistate System Subject to Imperfect Fault-Coverage

In this section, an algorithm for evaluating the reliability of a multistate system subject to imperfect coverage is proposed. Based on (12), the MSS with IPCM can be solved using the corresponding MSS under PCM.

$$\begin{aligned}&\text{System Reliability } y(R_s^I(t,h)) = \\ &\Pr\{\text{no uncovered failure in system}\}\times \\ &\Pr\{\text{system success} \mid \text{no uncovered failure in system}\},\end{aligned} \tag{13}$$

where

$$\begin{aligned}&\Pr\{\text{no uncovered failure in system}\}(P_u(t)) \\ &= \prod_{i\in S}\Pr\{\text{no uncovered failure of module } i\} \\ &= \prod_{i\in S}(1-P_i^I(t,0)) = \prod_{i\in S}R_i^I(t,1).\end{aligned} \tag{14}$$

Further,

$$R_i^c(t,j) = \Pr\{x_i^I \geq j \mid x_i^I > 0\} = R_i^I(t,j)/R_i^I(t,1). \tag{15}$$

This probability represents the conditional probability that module i under IPCM is in state j (i.e., performance level j) or above. Therefore, Pr{system success | no uncovered failures in system} can be obtained by substituting $R_i^c(t,j)$ for $R_i(t,j)$ in the corresponding PCM. Hence, the system reliability subject to imperfect coverage will be

$$R_s^I(t,h) = P_u(t)\times\left\{\begin{array}{l}\text{Reliability under PCM with}\\ R_i(t,j)=R_i^c(t,j), P_i(t,j)=P_i^c(t,j)\end{array}\right\}. \tag{16}$$

It should be noted that the same algorithm is applicable for availability evaluation, but, in this case, the input-set to the algorithm should be derived using component availability models. Moreover, the state probabilities of MSS under IPCM can be found as follows:

$$\begin{aligned}&P_s^I(t,h) = P_u(t)\times P_s^c(t,h)\\ &P_s^c(t,h) = P_s(t,h) \text{ of PCM with } P_i(t,j)=P_i^c(t,j)\\ &0\leq h\leq H,\ \ 0\leq j\leq m_i.\end{aligned} \tag{17}$$

Therefore, the procedure of the proposed algorithm is as follows:

1. Read the state probabilities of all modules, i.e., $P_i^I(t,j)$ for $i=1,\ldots,n$; $j=0,1,\ldots,m_i$.
2. For all i, find $R_i^I(t,1)\equiv 1-P_i^I(t,0)$.
3. Find $P_u(t)=\Pi_{i=1}^n R_i^I(t,1)$.
4. Find the conditional probability of each module at every level;
$$\begin{aligned}R_i^c(t,j) &= R_i^I(t,j)/R_i^I(t,1)\\ P_i^c(t,j) &= P_i^I(t,j)/R_i^I(t,1).\end{aligned} \tag{18}$$
5. Solve modular structures using the modularization method and deal with the dependency problem in the probability calculation using the OBDD method. Then, use these conditional probabilities to find the system reliability/availability (or the probability of a system state) at the required performance level h of the corresponding MSS under PCM and solve the generic problems as the following:
   - Conditional reliability $R_s^c(t,h)$: Find the conditional reliability of a MSS under PCM by substituting either $P_i^c(t,j)$ for $P_i(t,j)$ or $R_i^c(t,j)$ for $R_i(t,j)$.
   - Conditional availability $R_s^c(t,h)$: Find the conditional availability of a MSS under PCM by substituting either $P_i^c(t,j)$ for $P_i(t,j)$ or $R_i^c(t,j)$ for $R_i(t,j)$ (here, $R_i^c(t,j)$ and $R_i(t,j)$ are corresponding availabilities).
   - Conditional system state probability $P_s^c(t,h)$: Find the conditional probability of the state of a MSS with respect to PCM by substituting either $P_i^c(t,j)$ for $P_i(t,j)$ or $R_i^c(t,j)$ for $R_i(t,j)$.
6. Find the multistate system reliability/availability (or the probability of a system state) under IPCM using:
$$\begin{aligned}R_s^I(t,h) &= P_u(t)\times R_s^c(t,h)\\ P_s^I(t,h) &= P_u(t)\times P_s^c(t,h).\end{aligned} \tag{19}$$

Consider that, if a problem subject to imperfect fault-coverage is solved using a general multistate algorithm such as the Markov method, the computational time is proportional to $O(m+1)^{3n}$, where n is the number of modules in the system and each module has $m+1$ states including the imperfect coverage state. Furthermore, in this case, it is very difficult to combine with the modularization methods to solve the problem. In our method, only the m (instead of $m+1$) state probabilities are manipulated with the conditional probability method under imperfect fault-coverage model. Therefore, the percentage reduction (reduction factor) by using our method is approximately $1-(m/(m+1))^{3n}$. That means the reduction increases with n and decreases with m. However, this is a worst-case situation without any modular structure in a system. In our method, the advantage of using conditional probabilities makes it possible to apply this method for modular structures. If there exist some modular structures in the system, the modularization methods can be combined and n will be reduced. In general, our method is much faster than existing methods.

In Fig. 4, from the definition of MSS [6], the probability of event $x_{i:j}$ is

$$\Pr\{x_{i:j}\} = \sum_{k=j}^{m_i} P_i(t,k) = R_i(t,j). \tag{20}$$

For the calculation of the multistate system reliability, there exists dependency between $\Pr\{x_{3:4}\}$ and $\Pr\{x_{3:3}\}$ since "module 3 is operational at level 4 or above" implies "module 3 must be operational at level 3 or above." In Step 5 of our algorithm, we need not only to construct the OBDD representing the combination of performance requirements, but also to deal with the dependency problem in the probability calculation of the OBDD. The algorithm we adopt here is the Multistate Dependency Operation (MDO) algorithm in [18], whose idea is conceived from [38], and has a good result. And next, we combine our idea as described in Step 5 to solve the problem under IPCM. Fig. 5 shows the OBDD representations of three subrequirements of Fig. 4. The resultant OBDD and the probability calculation applied by MDO are shown as Fig. 6. It should be noted that $x_{3:4}$ is automatically eliminated during MDO. This is

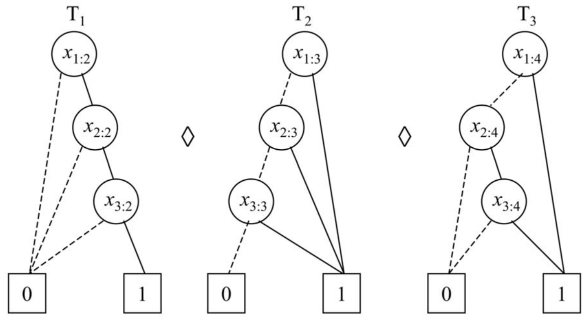


Fig. 5. The OBDD of subrequirement tree T_1, T_2, and T_3.

because, from the subrequirement T_2 in Fig. 4, when module 3 is operational in performance level 3 or above, it makes the system fit the requirement of being at performance level h or above. Hence, we don't need to consider if module 3 is operational in performance level 4 or above in subrequirement T_3, i.e., the node $x_{3:4}$ disappears. The detailed MDO operations and the probability calculations please refer to [18].

However, for the condition under IPCM in Step 5, the procedure of constructing a multistate OBDD is the same as the MDO, but the reliability calculation for IPCM should be made some modifications. We first find the conditional probability $\Pr\{x'_{i:j}\}$ of each multistate module i. Then, we use $\Pr\{x'_{i:j}\}$ instead of the probability or reliability of module i to calculate the conditional multistate system reliability R^c_{MSS}. Therefore, we combine our algorithm with the MDO and get the following: If $G = ite(x_{i:j}, G_1, G_0)$, $G_0 = ite(Z, H_1, H_0)$, and the order of node $x_{i:j}$ is smaller than that of node Z, the probability of G is

$$\Pr\{G\} = \begin{cases} \Pr\{x'_{i:j}\}[\Pr\{G_1\} - \Pr\{G_0\}] + \Pr\{G_0\} \\ \quad \text{(if } x_{i:j} \text{ and } Z \text{ belong to different modules)} \\ \Pr\{x'_{i:j}\}[\Pr\{G_1\} - \Pr\{H_1\}] + \Pr\{G_0\} \\ \quad \text{(if } x_{i:j} \text{ and } Z \text{ belong to the same module),} \end{cases} \tag{21}$$

where $\Pr\{x'_{i:j}\}$ is the conditional reliability of module i being operational at performance level j or above given that no uncovered failure occurred in that module (or module i).

Therefore, the probability of the OBDD's root node representing the conditional multistate system reliability $R^c_{MSS}(t,h)$ is obtained by (21). $P_u(t)$ is derived from (14). Hence, we get the multistate system reliability $R^I_{MSS}(t,h)$ under IPCM by

$$R^I_{MSS}(t,h) = P_u(t) \times R^c_{MSS}(t,h). \tag{22}$$

4 The Griffith's Importance Measure

In this section, we will first discuss the importance measure, also called the sensitivity analysis, of a multistate system under PCM. An OBDD-based method is proposed to evaluate the Griffith importance vector of a multistate system under PCM. With a little modification, the method is also applicable for computing the Griffith importance vector under imperfect fault-coverage model.

In [31], the Griffith importance vector, $\mathbf{I}^G_i$, was proposed to study how the particular states of a module contribute to a multistate system, and how the presence of a module and

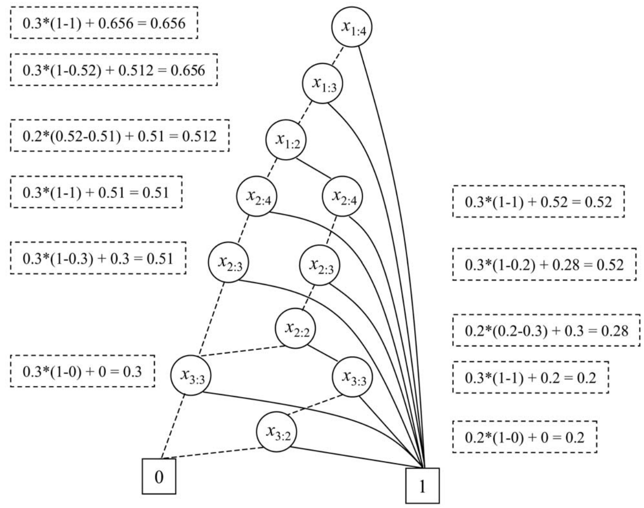


Fig. 6. The OBDD and the reliability evaluation of the system under PCM in Fig. 4.

a particular state of a module affect the contributions of other modules in the system:

$$\begin{aligned} \mathbf{I}^G(i) &= (I^G_1(i), \ldots, I^G_m(i)) \\ I^G_j(i) &= \sum_{h=1}^{H} (F_h - F_{h-1}) \cdot [R_{s|x_i=j}(t,h) - R_{s|x_i=j-1}(t,h)] \\ &= \sum_{h=1}^{H} (F_h - F_{h-1}) \cdot \nabla_{i:j}(R_s(t,h)), j = 1, \cdots, M_i, \end{aligned} \tag{23}$$

where F_h is the value of output/performance level when the multistate system is in state h, H is the index of maximum performance level of the system, $1 \le h \le H$, $R_{s|x_i=j}(t,h)$ is the reliability of a system being at performance level h or above given that module i is at performance level j, and $\nabla_{i:j}(R_s(t,h)) = R_{s|x_i=j}(t,h) - R_{s|x_i=j}(t,h)$.

The $I^G_j(i)$ in Griffith's importance vector can be interpreted as the range of the system performance when module i moves from state j to state $j-1$; it can be regarded as the importance of state j of system-module i. If the Boolean expression of $R_s(t,h)$ is transformed into the OBDD representation BDD_h as described in Section 3, then

$$\begin{aligned} \nabla_{i:j}(R_s(t,h)) &= \Pr\{BDD_{h|x_i=j}\} - \Pr\{BDD_{h|x_i=j-1}\} \\ &= \Pr\{BDD_{h|(x_{i:1},\cdots,x_{i:j})=1,(x_{i:j+1},\cdots,x_{i:m_i})=0}\} \\ &\quad - \Pr\{BDD_{h|(x_{i:j},\cdots,x_{i:j-1})=1,(x_{i:j},\cdots,x_{i:m_i})=0}\}. \end{aligned} \tag{24}$$

4.1 Two-Pass Traversal Method

This method needs to traverse the OBDD twice to obtain $\nabla_{i:j}(R_s(t,h))$.

- Find

  $$\Pr\{BDD_{h|x_i=j}\} = \Pr\{BDD_{h|(x_{i1},\cdots,x_{i:j})=1,(x_{i:j+1},\cdots,x_{i:m_i})=0}\}$$

  by assuming that module i is running at level j, i.e., $(x_{i:1},\cdots,x_{i:j}) = 1$, $(x_{i:j+1},\cdots,x_{i:m_i}) = 0$.
- Find

  $$\Pr\{BDD_{h|x_i=j-1}\} = \Pr\{BDD_{h|(x_{i:1},\cdots,x_{i:j-1})=1,(x_{i:j},\cdots,x_{i:m_i})=0}\}$$

by assuming that module i is running at level $j-1$, i.e., $(x_{i:1},\cdots,x_{i:j-1})=1$, $(x_{i:j},\cdots,x_{i:m_i})=0$.

Then, $\nabla_{i:j}(R_s(t,h)) = \Pr\{BDD_{h|x_i=j}\} - \Pr\{BDD_{h|x_i=j-1}\}$ can be obtained and, therefore, we can derive the Griffith's importance vector by (23) and (24).

4.2 Single-Pass Traversal Method

This method needs to traverse the OBDD only once to get $\nabla_{i:j}(R_s(t,h))$. There are two parts. First, to evaluate $\nabla_{i:j}(R_s(t,h))$ from (24), we found that the only difference between computing $\Pr\{BDD_{h|x_i=j}\}$ and $\Pr\{BDD_{h|x_i=j-1}\}$ is to let $x_{i:j}=1$ and $x_{i:j}=0$ in the OBDD traversal, respectively. Therefore, only the disjoint path of the OBDD that goes from the root to the terminal one and does include node $x_{i:j}$ in it will contribute to the importance measure. We should delete the disjoint paths that do not include node $x_{i:j}$ or let the probabilities of those paths be 0 during the OBDD traversal.

In the second part, we can combine the two calculations of Section 4.1 at node $x_{i:j}$. When we visit node $x_{i:j}$ in the OBDD traversal, the probability of the node is equivalent to the probability of the right subtree (i.e., $x_{i:j}=1$) minus the probability of the left subtree (i.e., $x_{i:j}=0$). Further, let $(x_{i:1},\cdots,x_{i:j-1})=1$ and $(x_{i:j+1},\cdots,x_{i:m_i})=0$ since module i moves only from state j to state $j-1$ and does not affect the other states. If the Boolean expression of OBDD at node v is $f_v = ite(v, f_{v=1}, f_{v=0})$, $\Pr\{v\}$ is the probability of the Boolean variable, where node v corresponds, being true, node u is the subnode of node v, and the Boolean expression of OBDD at node u is $f_u = ite(u, f_{u=1}, f_{u=0})$, then the single-pass traversal algorithm to evaluate $\nabla_{i:j}(R_s(t,h))$ from (24) is as following:

Compute the probability $\Pr\{f_v\}$ of each node (say v) from the bottom to the up of OBDD using the following steps:

1. If node v has been visited, then retrieve $\Pr\{f_v\}$ from the computed table and go to Step 8.
2. If node v corresponds to $x_{i:1},\cdots,x_{i:j-1}$, then let $\Pr\{v\}=1$ (i.e., $(x_{i:1},\cdots,x_{i:j-1})=1$). Go to Step 5.
3. If node v corresponds to $x_{i:j+1},\cdots,x_{i:m_i}$, then let $\Pr\{v\}=0$ (i.e., $(x_{i:j+1},\cdots,x_{i:m_i})=0$). Go to Step 6.
4. If node v corresponds to $x_{i:j}$, then

$$\Pr\{f_v\} = \begin{cases} \Pr\{f_{v=1}\} - \Pr\{f_{v=0}\} & \text{(if } v \text{ and } u \text{ belong to different modules)} \\ \Pr\{f_{v=1}\} - \Pr\{f_{u=1}\} & \text{(if } v \text{ and } u \text{ belong to the same module).} \end{cases}$$

5. If node v does not correspond to $x_{i:j}$ and $\text{ordering}(v) > \text{ordering}(x_{i:j})$, then

$$\Pr\{f_v\} = \begin{cases} \Pr\{v\}[\Pr\{f_{v=1}\} - \Pr\{f_{v=0}\}] + \Pr\{f_{v=0}\} \\ \quad \text{(if } v \text{ and } u \text{ belong to different modules)} \\ \Pr\{v\}[\Pr\{f_{v=1}\} - \Pr\{f_{u=1}\}] + \Pr\{f_{v=0}\} \\ \quad \text{(if } v \text{ and } u \text{ belong to the same module).} \end{cases} \tag{25}$$

6. If node v does not correspond to $x_{i:j}$ and $\text{ordering}(v) < \text{ordering}(x_{i:j})$, then also use (25) to calculate $\Pr\{f_v\}$ except that let $\Pr\{f_{v=1}\}$ or $\Pr\{f_{v=0}\}$ be 0 in (25) if the right or left subtree is independent of $x_{i:j}$, respectively. To check if the subtree of node v is independent of $x_{i:j}$ is simple. If $\text{ordering}(u) > \text{ordering}(x_{i:j})$, then the subtree is independent of $x_{i:j}$.
7. Record node v and $\Pr\{f_v\}$ into the computed table.
8. Visit next node.

Therefore, the probability $\Pr\{f_v\}$ of the root node (top node) of the OBDD represents $\nabla_{i:j}(R_s(t,h))$ and can be efficiently obtained by the single-pass OBDD traversal. The pseudo code of the algorithm for computing the Griffith's importance measure is shown in Fig. 7. It should be noted that the computed table recording the probabilities of the OBDD nodes is also used to avoid repeated computations. This scheme improves the efficiency of the algorithm.

```
Main() {
    GIM(BDDnode root);
}

Procedure double GIM(BDDnode v) {        // Griffith's Importance Measure
    double result, fv_1, fv_0, fu_1;
    BDDnode u;
    if (v = BDD_one ) then return 1;
    if (v = BDD_zero ) then return 0;
    if ( result = get_computed_table(v) is a hit ) then return (result);

    fv_1 = GIM( sub_node_true(v) );
    fv_0 = GIM( sub_node_false(v) );
    u = sub_node_false(v);
    fu_1 = GIM( sub_node_true(u) );

    if (v is corresponding to x_{i:1}, ..., x_{i:j-1}) then Pr{v}=1;
    if (v is corresponding to x_{i:j+1}, ..., x_{i:mi}) then Pr{v}=0;
    if (v is corresponding to x_{i:j}) then
        if (v and u belong to different modules) then
            result = fv_1 – fv_0;
        else
            result = fv_1 – fu_1;
        end if
    else if ( ordering(v) > ordering(x_{i:j}) ) then
        if (v and u belong to different modules) then
            result = Pr{v} * (fv_1 – fv_0) + fv_0;
        else
            result = Pr{v} * (fv_1 – fu_1) + fv_0;
        end if
    else if ( ordering(v) < ordering(x_{i:j}) ) then
        if sub_node_true(v) is independent of x_{i:j} then fv_1 = 0;
        if sub_node_false(v) is independent of x_{i:j} then fv_0 = 0;
        if (v and u belong to different modules) then
            result = Pr{v} * (fv_1 – fv_0) + fv_0;
        else
            result = Pr{v} * (fv_1 – fu_1) + fv_0;
        end if
    end if
    insert_computed_table(v, result);
    return (result);
}
```

Fig. 7. The pseudo code of the algorithm for computing the Griffith's importance measure.

4.3 Griffith's Importance Measure under Imperfect Coverage Model

In this section, we will discuss the Griffith's importance measure under imperfect coverage model and propose a method to compute $\nabla_{i:j}R_s^I(t,h))$. For the imperfect fault-coverage model, from (14), the probability of no uncovered failure in the system is

$$P_u(t) = \prod_{v=1}^{n} R_v^I(t,1) \tag{26}$$

and the conditional state probability of module i given that module i is not in failed and uncovered state is

$$P_i^c(t,j) = \frac{P_i^I(t,j)}{P_i^I(t,1) + \cdots + P_i^I(t,m_i)}. \tag{27}$$

For evaluating the Griffith's importance measure under the imperfect coverage model, by (16), we have

$$\begin{aligned}\nabla_{i:j}(R_s^I(t,h)) &= \nabla_{i:j}(P_u(t) \times R_s^c(t,h)) \\ &= R_s^c(t,h)\nabla_{i:j}(P_u(t)) + P_u(t)\nabla_{i:j}(R_s^c(t,h)),\end{aligned} \tag{28}$$

where

$$\begin{aligned}\nabla_{i:j}(P_u(t)) = \prod_{v=1, v\neq i}^{n} & R_v^I(t,1) \cdot [P_i^I(t,1) + \cdots + P_i^I(t,j-1) \\ & + P_i^I(t,j+1) + \cdots + P_i^I(t,m_i)]\end{aligned} \tag{29}$$

and, by the chain rule of differentiation,

$$\begin{aligned}\nabla_{i:j}\big(R_s^c(t,h)\big) &= \frac{\partial R_s^c(t,h)}{\partial P_i^c(t,j)} \cdot \frac{\partial P_i^c(t,j)}{\partial P_i^I(t,j)} \\ \frac{\partial P_i^c(t,j)}{\partial P_i^I(t,j)} &= \frac{(P_i^I(t,1) + \cdots + P_i^I(t,m_i)) + P_i^I(t,j)}{(P_i^I(t,1) + \cdots + P_i^I(t,m_i))^2} \\ &= \frac{1}{P_i^I(t,1) + \cdots + P_i^I(t,m_i)}[1 + P_i^c(t,j)] \\ &= \frac{1}{R_i^I(t,1)}[1 + P_i^c(t,j)].\end{aligned} \tag{30}$$

Therefore,

$$\begin{aligned}\nabla_{i:j}(R_s^I(t,h)) = R_s^c(t,h) \prod_{\substack{v=1 \\ v\neq i}}^{n} & R_v^I(t,1) \cdot [P_i^I(t,1) + \cdots + P_i^I(t,j-1) \\ & + P_i^I(t,j+1) + \cdots + P_i^I(t,m_i)] \\ & + \frac{P_u(t)}{R_i^I(t,1)}[1 + P_i^c(t,j)] \cdot \frac{\partial R_s^c(t,h)}{\partial P_i^c(t,j)}.\end{aligned} \tag{31}$$

In order to derive $\nabla_{i:j}(R_s^I(t,h))$, we need to find only $\partial R_s^c(t,h)/\partial P_i^c(t,j)$ because all the parameters are known already. Finding $\partial R_s^c(t,h)/\partial P_i^c(t,j)$ is equivalent to finding $\nabla_{i:j}(R_s(t,h))$ under PCM as described in Section 4.2, but with the modified values of the module's state reliabilities, that is, the conditional state reliability of module i, $P_i^c(t,j)$. Then, $\nabla_{i:j}(R_s^I(t,h))$ can be evaluated and, therefore, the Griffith's importance vectors under IPCM can be derived.

5 Illustrative Examples

Example 1. Let us consider an example of Fault-Tolerant Parallel Processor (FTPP) in [39]. Fig. 8 is an instance of an FTPP cluster that consists of 16 processing elements (PEs), with four connected to each of four network elements (NEs). This configuration divides the active elements of a triad among NE1, NE2, and NE3, and uses the PEs on NE4 as spares. The PEs that are in the same relative position on first 3 NEs form a triad, and the PE in the same relative position on NE4 serves as a hot spare for the triad.

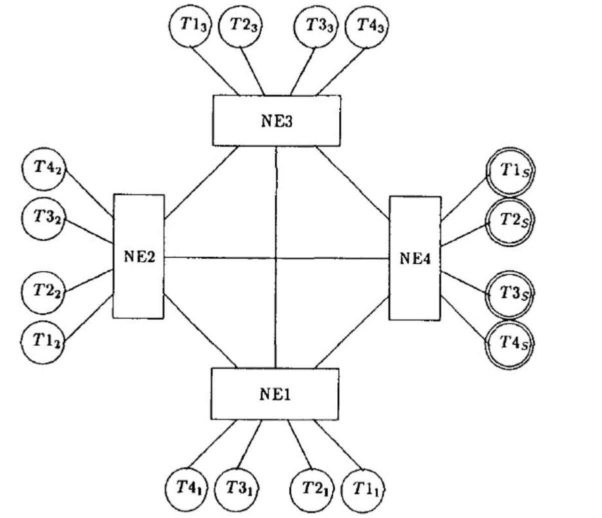


Fig. 8. An FTPP cluster with one spare per triad.

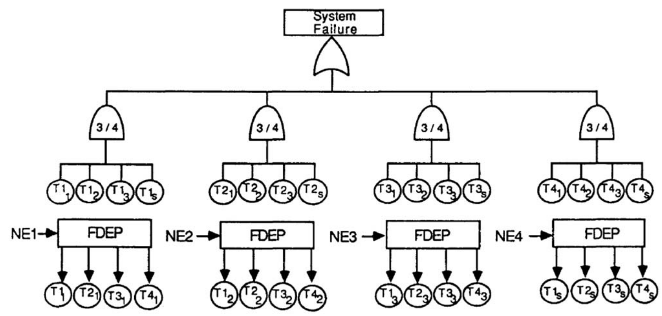


Fig. 9. Fault-Tree Model for Fig. 8.

The fault-tree model (Fig. 9) for this configuration uses four functional-dependency gates (FDEP) [39] to reflect the dependence of the PEs on the NEs. The FDEP has one trigger-input and one or more dependent basic events. In an FDEP, the dependent basic events are functionally dependent on the trigger event. When the trigger event occurs, the dependent basic events are forced to occur. The FDEP gates are not explicitly connected to the other gates in the tree since the reliability requirements (all four triads must be operational) do not explicitly mention the NEs. Fig. 9 shows four 3/4 gates connected to the top OR gate, one 3/4 gate for each triad. A triad fails when only one member PE remains (three of the four PEs in the triad have failed).

Let us consider this FTPP as a multistate system subject to imperfect coverage. NE1 is a trigger event for an FDEP gate. System fails if NE1 fails in uncovered mode. If its failure is covered, then all components connected to NE1 do not work properly in the FTPP cluster. In addition, if NE1 is not failed, then the components connected to NE1 may acts independently. It means that they can fail in covered or uncovered mode or can function properly. Obviously, the system will be in better performance if more NEs are functioning.

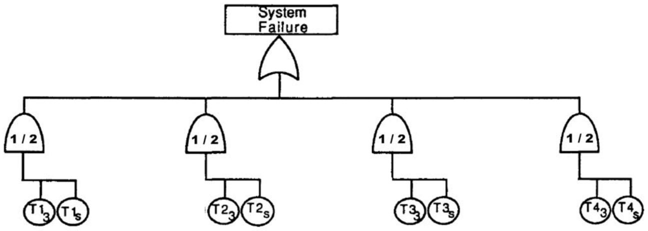


Fig. 10. The failure logic of system running in performance state 11.

Therefore, if we consider NE1, NE2, NE3, and NE4 as the key roles of system performance, there will be $2^4 + 1 = 17$ states:

state 16: all NEs are good. $P_s^I(t,16) = (p_1 \cdot p_2 \cdot p_3 \cdot p_4) \times (1 - U(t,16))$

state 15: NE1 failure is covered—others are good. $P_s^I(t,15) = ((q_1 c_1) \cdot p_2 \cdot p_3 \cdot p_4) \times (1 - U(t,15))$.

⋮

state 11: NE1 and NE2 failed and covered—others are good. $P_s^I(t,11) = ((q_1 c_1) \cdot (q_2 c_1) \cdot p_3 \cdot p_4) \times (1 - U(t,11))$.

⋮

state 1: all failed and covered. $P_s^I(t,1) = ((q_1 c_1) \cdot (q_2 c_2) \cdot (q_3 c_3) \cdot (q_4 c_4)) \times (1 - U(t,1))$.

state 0: at least one uncovered failure. $P_s^I(t,0) = 1 - (p_1 + q_1 c_1) \cdot (p_2 + q_2 c_2) \cdot (p_3 + q_3 c_3) \cdot (p_4 + q_4 c_4)$,

where p_i, q_i, and c_i are the reliability, unreliability, and coverage factor of NEi, respectively. $U(t,h)$ is the unreliability of the system configuration at performance state h (Note: Different performance state has different system configuration). Then, the multistate system reliability subject to IPCM is $R_s^I(t,h) = \sum_{k=h}^{16} P_s^I(t,k)$.

The imperfect coverage of NEs introduces the temporal system failure logic. Therefore, the sequence of events is important and it has to be solved using Markov chains. However, the computation becomes very complex if there exist multiple states for each component or each PE including failed and uncovered state. On the contrary, using the concepts of our approach, we can solve it using combinatorial models easily. For example, if the required performance level of the system is in state 11 and each element has three states (*not failed*, *failed and covered*, and *failed and uncovered*), the failure logic of this con-figuration is shown in Fig. 10.

To evaluate the system unreliability of this configuration (or this performance state) under IPCM, first we compute the conditional probability of each element by (18). Using the conditional probability of each element, the conditional system unreliability can be derived by Step 5. Therefore, the probability of system unreliability in performance state 11 subject to imperfect fault-coverage is obtained by (19). Generally, our method can be easily combined with modularization techniques to obtain the reliability or unreliability of different performance states under IPCM. The overall multistate system reliabilities $R_s^I(t,h)$ are described in Table 1.

TABLE 1
The Multistate System Reliability of the FTPP Example

Unreliability		Coverage		Unreliability (coverage = 0.9 for all elements)							
NE1(q_1)	0.15	NE1(c_1)	0.75	$T1_1$	$T2_1$	$T3_1$	$T4_1$	0.15	0.18	0.10	0.04
NE2(q_2)	0.18	NE2(c_2)	0.90	$T1_2$	$T2_2$	$T3_2$	$T4_2$	0.17	0.20	0.09	0.05
NE3(q_3)	0.21	NE3(c_3)	0.85	$T1_3$	$T2_3$	$T3_3$	$T4_3$	0.13	0.21	0.12	0.05
NE4(q_4)	0.16	NE4(c_4)	0.95	$T1_S$	$T2_S$	$T3_S$	$T4_S$	0.14	0.16	0.09	0.03

State h	16	15	14	13	12	11	10	9
Pr{h}	0.4625	0.0612	0.0914	0.1045	0.0837	0.0121	0.0138	0.0111
$U(t,h)$	0.0373	0.1747	0.1661	0.1679	0.1853	0.6336	0.6340	0.6702
$P_s^I(t,h)$	0.4453	0.0505	0.0762	0.0870	0.0682	0.0044	0.0051	0.0037
$R_s^I(t,h)$	0.4453	0.4958	0.5720	0.6590	0.7271	0.7316	0.7366	0.7403
State h	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
Pr{h}	0.0206	0.0165	0.0189	0.0027	0.0022	0.0025	0.0037	0.0005
$U(t,h)$	0.6160	0.6540	0.6543	1.0000	1.0000	1.0000	1.0000	1.0000
$P_s^I(t,h)$	0.0079	0.0057	0.0065	0.0000	0.0000	0.0000	0.0000	0.0000
$R_s^I(t,h)$	0.7482	0.7539	0.7605	0.7605	0.7605	0.7605	0.7605	0.7605

Example 2. Let us consider a simple bridge network shown in Fig. 11a. The two-state OBDD-based path function of this network system is

$$F = x_1x_3x_5 + x_1x_4 + x_2x_5 + x_2x_5 + x_2x_3x_4$$

and is constructed by the algorithm in [16] as shown in Fig. 11b. Let us consider the multistate network system. Assume that the redundancy techniques are used so that each link has a fault-tolerance scheme. Therefore, we can treat a link as a module with various link capacities or with various performance levels (i.e., a multistate network system) and the fault-coverage condition should be considered.

Case 1. The basic requirement for the system being in an acceptable performance level is:

$$\Phi_{accept} = x_{1:2}x_{3:2}x_{5:2} + x_{1:2}x_{4:2} + x_{2:2}x_{5:2} + x_{2:2}x_{3:2}x_{4:2}.$$

Case 2. The path x_1x_4 is the backbone of the network. Most of the dataflow run through the path and, thus, the limitation of the requirement for the path is much more strict. Therefore, if x_1 needs to be at least in level 5 and x_4

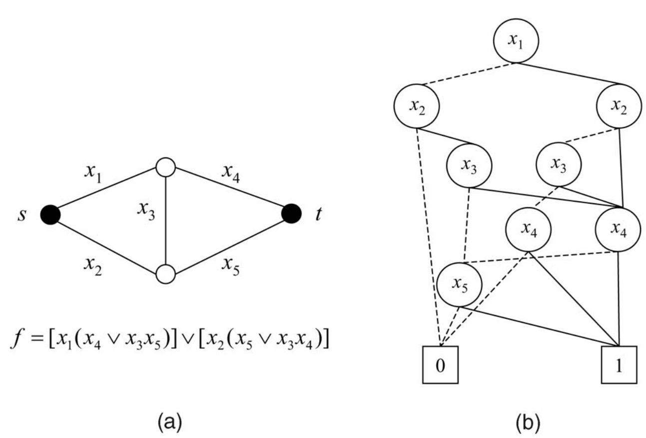
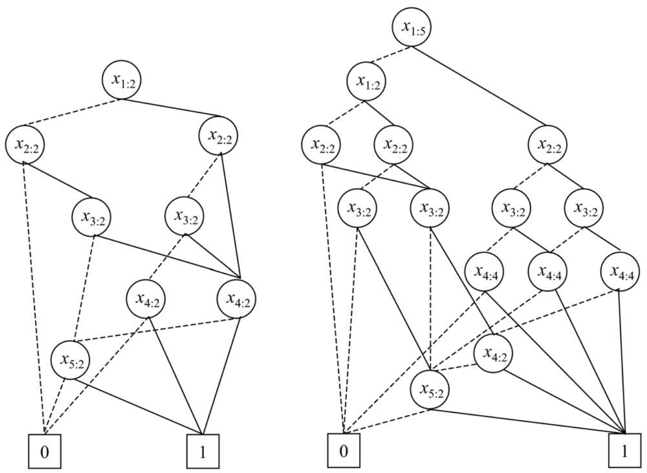


Fig. 11. (a) A bridge network. (b) The corresponding OBDD-based path function.



(a) (b)

Fig. 12. The resultant OBDD after MDO. (a) Φ_{accept}. (b) Φ_{good}.

needs to be at least in level 4 for path x_1x_4, the requirement for the system being in a good performance level is:

$$\Phi_{good} = x_{1:2}x_{3:2}x_{5:2} + x_{1:5}x_{4:4} + x_{2:2}x_{5:2} + x_{2:2}x_{3:2}x_{4:2}.$$

Fig. 12 shows the results of Φ_{accept} and Φ_{good} after applying MDO. Table 2 illustrates the parameters of a module with the assumption that each module has six performance states including failed and uncovered state. If all modules are identical, the system reliabilities of Φ_{accept} and Φ_{good} with different coverage factors are obtained by (21) and (22) as shown in Table 3.

Table 4 illustrates the $\nabla_{i:j}(R_s(t,h))$ and $\nabla_{i:j}(R_s^I(t,h))$ of Φ_{accept} and Φ_{good} obtained by our algorithm under PCM and IPCM with different coverage factors. The result shows that the importance measure of each module becomes larger if imperfect fault-coverage is taken into consideration. For Φ_{good}, $x_{1:2}$ is more important than $x_{1:5}$. This is because that x_1x_4 is the backbone and, if $x_{1:5}$ is not functioning, we can use an alternative path to slightly compensate the dataflow. However, if the performance of x_1 drops too much (below $x_{1:2}$), the system cannot afford it. Therefore, maintaining $x_{1:2}$ to be functional is more important than $x_{1:5}$. If we apply the

TABLE 3
Reliability with a Coverage Factor of $c_1 > c_2$

	Perfect	Imperfect	
		c_1	c_2
Φ_{accept}	0.97848	0.57472	0.31654
Φ_{good}	0.95692	0.55947	0.30642

Markov techniques or other techniques on each module to solve the behavior of state's transition, then the time-dependent multistate system reliability of different performance levels with different coverage factors is shown in Fig. 13. Fig. 13 shows that Φ_{good} is less reliable than Φ_{accept}. That means we need to pay more on the system if we want to increase the reliability of Φ_{good} to be the same as that of Φ_{accept}.

6 Conclusions

This paper has proposed a model for multistate systems with imperfect fault-coverage. An OBDD-based approach for the evaluation of the multistate system reliability and the Griffith's importance measure has also been proposed. With a little modification, the OBDD-based method can be applied to the multistate systems under IPCM. The evaluation of the system reliability and importance measure is very helpful for the optimal design of a multistate system. It was shown that with the application of conditional probabilities, the time complexity of this method for reliability evaluation is equivalent to that of the methods for perfect coverage model. Furthermore, the approach used to evaluate the reliability can also be employed to evaluate the availability of a system if the Markov process is applied to analyze the behavior of the state transition of each module.

In [6] and [40], with helpful comments from [6], they have presented various performance measures related to multistate systems. In order to compute these measures, we need to find the reliability of a system at various performance levels. Therefore, the results of this paper can be integrated to find the performance measures of a multistate system. This process is straightforward and, therefore, it is not discussed here explicitly [6]. Also, the method proposed for IPCM in this paper can be integrated

TABLE 2
The Parameters of Individual Module with a Coverage Factor of $c_1 > c_2$

	Perfect			Imperfect					
				c_1			c_2		
j	$P_i^I(t,j)$	$P_i^c(t,j)$	$R_i^c(t,j)$	$P_i^I(t,j)$	$P_i^c(t,j)$	$R_i^c(t,j)$	$P_i^I(t,j)$	$P_i^c(t,j)$	$R_i^c(t,j)$
0	--	--	--	0.1	--	--	0.2	--	--
1	0.1	0.1	1	0.1	0.1111	1.0000	0.1	0.1250	1.0000
2	0.2	0.2	0.9	0.2	0.2222	0.8889	0.2	0.2500	0.8750
3	0.3	0.3	0.7	0.3	0.3333	0.6667	0.2	0.2500	0.6250
4	0.3	0.3	0.4	0.2	0.2222	0.3333	0.2	0.2500	0.3750
5	0.1	0.1	0.1	0.1	0.1111	0.1111	0.1	0.1250	0.1250

TABLE 4
Importance Measure with a Coverage Factor of $c_1 > c_2$

		Φ_{accept}			Φ_{good}		
		$\nabla_{i:j}(R_s(t,h))$	$\nabla_{i:j}(R_s^I(t,h))$		$\nabla_{i:j}(R_s(t,h))$	$\nabla_{i:j}(R_s^I(t,h))$	
i	j		c_1	c_2		c_1	c_2
1	2	0.1062	0.3588	0.1460	0.0810	0.3273	0.1243
2	2	0.1062	0.3588	0.1460	0.2448	0.4711	0.2241
3	2	0.0162	0.2796	0.0900	0.1548	0.3919	0.1681
4	2	0.1062	0.3588	0.1460	0.0810	0.3273	0.1243
5	2	0.1062	0.3588	0.1460	0.2448	0.4711	0.2241
1	5	--	--	--	0.0112	0.3020	0.0953
4	4	--	--	--	0.0838	0.3304	0.1271

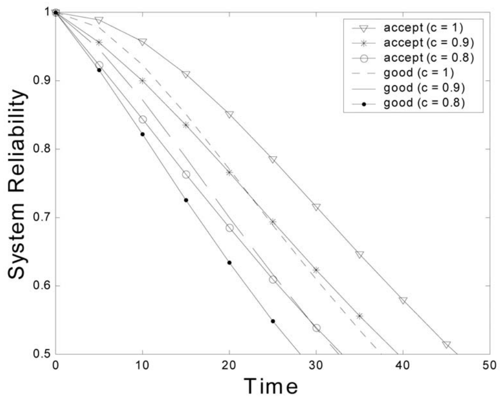


Fig. 13. The system reliability of Φ_{accept} and Φ_{good} with different coverage factors.

with the technique [13], [15], [16] that uses OBDD for reliability analysis of a system.

For complex systems such as fault-tolerant computer systems, network systems with variable link-capacities, and so on, this approach is applicable. It generates the complete results more quickly and accurately even when there exist a number of dependencies such as shared loads (reconfiguration), degradation, common-cause failures, and so on. Based on this approach, researches on failure frequency analysis and optimal design issues of a multistate system will be the focus of our future works.

Acknowledgments

This research was supported by the National Science Council, Taiwan, R.O.C. under grant NSC 92-2213-E-002-007.

References

[1] J.D. Murchland, "Fundamental Concepts and Relations for Reliability Analysis of Multistate Systems," *Reliability and Fault Tree Analysis, Theoretical, and Applied Aspects of System Reliability (SIAM),* pp. 581-618, 1975.

[2] A.P. Wood, "Multistate Block Diagrams and Fault Trees," *IEEE Trans. Reliability,* vol. 34, no. 3, pp. 236-240, Aug. 1985.

[3] S. Garribba, E. Guagnini, and P. Mussio, "Multiple-Valued Logic Trees: Meaning and Prime Implicants," *IEEE Trans. Reliability,* vol. 34, no. 5, pp. 463-472, Dec. 1985.

[4] A. Gandini, "Importance and Sensitivity Analysis in Assessing System Reliability," *IEEE Trans. Reliability,* vol. 39, no. 1, pp. 61-69, Apr. 1990.

[5] J. Xue, "On Multistate System Analysis," *IEEE Trans. Reliability,* vol. 34, no. 4, pp. 329-337, Oct. 1985.

[6] J. Xue and K. Yang, "Dynamic Reliability Analysis of Coherent Multistate Systems," *IEEE Trans. Reliability,* vol. 44, pp. 683-688, Dec. 1995.

[7] G. Levitin, "Incorporating Common-Cause Failure into Nonrepairable Multistate Series-Parallel System Analysis," *IEEE Trans. Reliability,* vol. 50, pp. 380-388, Dec. 2001.

[8] J.B. Dugan, "Fault Tree and Imperfect Coverage," *IEEE Trans. Reliability,* vol. 38, pp. 177-185, June 1989.

[9] S.V. Amari, J.B. Dugan, and R.B. Misra, "Optimal Reliability of Systems Subject to Imperfect Fault-Coverage," *IEEE Trans. Reliability,* vol. 48, pp. 275-284, Sept. 1999.

[10] A. Reibman and K.S. Trivedi, "Numerical Transient Analysis of Markov Models," *Computers and Operations Research,* vol. 15, no. 1, pp. 19-36, 1998.

[11] K.S. Trivedi, *Probability and Statistics with Reliability, Queuing, and Computer Science Applications.* Prentice-Hall, 1982.

[12] R.E. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation," *IEEE Trans. Computers,* vol. 35, no. 8, pp. 677-691, Aug. 1986.

[13] Y.R. Chang, S.V. Amari, and S.Y. Kuo, "Computing System Failure Frequencies and Reliability Importance Measure Using OBDD," *IEEE Trans. Computers,* vol. 53, no. 1, pp. 54-68, Jan. 2004.

[14] Y.R. Chang, S.V. Amari, and S.Y. Kuo, "Reliability Evaluation of Multi-State Systems Subject to Imperfect Coverage Using OBDD," *Proc. 2002 Pacific Rim Int'l Symp. Dependable Computing (PRDC '02),* pp. 193-200, 2002.

[15] F.M. Yeh, S.K. Lu, and S.Y. Kuo, "OBDD-Based Evaluation of *K*-Terminal Network Reliability," *IEEE Trans. Reliability,* vol. 51, no. 4, pp. 443-451, Dec. 2002.

[16] S.Y. Kuo, S.K. Lu, and F.M. Yeh, "Determining Terminal-Pair Reliability Based on Edge Expansion Diagrams Using OBDD," *IEEE Trans. Reliability,* vol. 48, pp. 234-246, Sept. 1999.

[17] X. Zang, H. Sun, and K.S. Trivedi, "Dependability Analysis of Distributed Computer Sys-tems with Imperfect Coverage," *Proc. 29th Ann. Int'l Symp. Fault-Tolerant Computing (FTC-29),* pp. 330-337, 1999.

[18] X. Zang, D. Wang, H. Sun, K.S. Trivedi, "A BDD-Based Algorithm for Analysis of Multistate Systems with Multistate Components," *IEEE Trans. Computers,* vol. 52, no. 12, pp. 1608-1618, Dec. 2003.

[19] B. Lin, O. Coudert, and J.C. Madre, "Symbolic Prime Generation for Multiple-Valued Function," *Proc. 29th ACM/IEEE Design Automation Conf.,* pp. 40-44, 1992.

[20] E.A. Elsayed, *Reliability Engineering.* Addison Wesley Longman, 1996.

[21] A. Hoyland and M. Rausand, *System Reliability Theory: Models and Statistical Methods.* John Wiley & Sons, 1994.

[22] F.C. Meng, "Comparing the Importance of System Components by Some Structural Characteristics," *IEEE Trans. Reliability,* vol. 45, no. 1, pp. 59-65, Mar. 1996.

[23] F.C. Meng, "Some Further Results on Ranking the Importance of System Components," *Reliability Eng. and System Safety,* vol. 47, pp. 97-101, 1995.

[24] J.S. Hong and C.H. Lie, "Joint Reliability-Importance of Two Edges in an Undirected Network," *IEEE Trans. Reliability,* vol. 42, no. 1, pp. 17-23, Mar. 1993.

[25] M.J. Armstrong, "Joint Reliability-Importance of Components," *IEEE Trans. Reliability,* vol. 44, no. 3, pp. 408-412, Sept. 1995.

[26] R.E. Barlow and A.S. Wu, "Coherent Systems with Multistate Components," *Math. Operations Research,* vol. 3, pp. 275-281, 1978.

[27] T. Aven, "On Performance Measures for Multistate Monotone Systems," *Reliability Eng. and System Safety,* vol. 41, pp. 259-266, 1993.

[28] B. Natvig, "Two Suggestions of How to Define a Multistate Coherent System," *Advanced Applied Probability,* vol. 14, pp. 434-457, 1982.

[29] H.W. Block, "A Decomposition for Multistate Monotone System," *J. Applied Probability,* vol. 19, pp. 391-402, 1982.

[30] F.C. Meng, "Component-Relevancy and Characterization Results in Multistate Systems," *IEEE Trans. Reliability,* vol. 42, no. 3, pp. 478-483, Sept. 1993.

[31] W.S. Griffith, "Multistate Reliability Models," *J. Applied Probability,* vol. 17, pp. 735-744, 1980.

[32] S. Wu and L.Y. Chan, "Performance Utility-Analysis of Multistate Systems," *IEEE Trans. Reliability,* vol. 52, no. 1, pp. 14-21, Mar. 2003.

[33] A. Rauzy, "New Algorithms for Fault Tree Analysis," *Reliability Eng. and System Safety,* vol. 40, pp. 203-211, 1993.

[34] R.M. Sinnamon and J.D. Andrews, "Improved Efficiency in Qualitative Fault Tree Analysis," *Quality and Reliability Eng. Int'l,* vol. 13, pp. 293-298, 1997.

[35] S.V. Amari, J.B. Dugan, and R.B. Misra, "A Separable Method for Incorporating Imperfect Fault-Coverage into Combinatorial Models," *IEEE Trans. Reliability,* pp. 267-274, Sept. 1999.

[36] J.B. Dugan and K.S. Trivedi, "Coverage Modeling for Dependability Analysis of Fault-Coverage Systems," *IEEE Trans. Computers,* vol. 38, pp. 775-787, June 1989.

[37] S.A. Doyle, J.B. Dugan, and F.A. Patterson-Hine, "A Combinatorial Approach to Modeling Imperfect Coverage," *IEEE Trans. Reliability,* vol. 44, pp. 87-94, Mar. 1995.

[38] H. Sun, X. Zang, and K.S. Trivedi, "A BDD-Based Algorithm for Reliability Analysis of Phased-Mission Systems," *IEEE Trans. Reliability,* vol. 48, no. 1, pp. 50-60, Mar. 1999.

[39] J.B. Dugan, S.J. Bavuso, and M.A. Boyd, "Dynamic Fault-Tree Models for Fault-Tolerant Computer Systems," *IEEE Trans. Reliability,* vol. 41, no. 3, pp. 363-377, Sept. 1992.
[40] S.V. Amari and R.B. Misra, "Comment on: Dynamic Reliability Analysis of Coherent Multi-State Systems," *IEEE Trans. Reliability,* vol. 46, pp. 460-461, Dec. 1997.

Yung-Ruei Chang received the MS degree (1995) and the PhD degree (2004) in electrical engineering from National Taiwan University. He is an assistant researcher at the Institute of Nuclear Energy Research (INER), Atomic Energy Council, Taiwan, where he has been working since 1996. He is a member of INER's fourth nuclear power plant development team and is involved in system reliability and fault-tolerant system design. His research interests include nuclear system reliability analysis, fault-tolerant system, dependable computing, and software reliability. He is a member of the IEEE.

Suprasad V. Amari received the BS degree (1990) in mechanical engineering from Sri Venkateswara University, Tirupati, the MS degree in reliability engineering (1992) and the PhD degree in reliability, risk, and fault tolerance of complex systems (1998) from the Indian Institute of Technology, Kharagpur. He is a senior reliability engineer at Relex Software Corporation. He is a member of Relex development team and consulting services. He was with Tata Consultancy Services from 1996 to 2000, where he was involved in software design and development using objected-oriented methodologies and formal methods, and worked as a consultant and technical lead for data mining and data warehousing projects. His research interests include hardware and software reliability, risk assessment, fault-tolerant computing, and optimization. He has published more than 30 research papers in reputed international journals and conferences. He is an editorial board member of the *International Journal of Reliability*, *Quality and Safety Engineering*, and the *International Journal on Performability Engineering*. He also is a management committee member of RAMS, advisory board member of several international conferences, and a reviewer for several journals on reliability and safety. He is a senior member of ASQ and the IEEE and a member of IIE, ACM, ASA, SSC, SRE, and SOLE.

Sy-Yen Kuo received the BS degree (1979) in electrical engineering from National Taiwan University, the MS degree (1982) in electrical and computer engineering from the University of California at Santa Barbara, and the PhD degree (1987) in computer science from the University of Illinois at Urbana-Champaign. He is the dean of the College of Electrical Engineering and Computer Science, National Taiwan University, Keelung, Taiwan. He is also a professor in the Department of Electrical Engineering, National Taiwan University where he is currently on leave and was the Chairman of the same department from 2001 to 2004. He spent his sabbatical years as a visiting professor in the Computer Science and Engineering Department at the Chinese University of Hong Kong from 2004-2005 and as a visiting researcher at AT&T Labs-Research, New Jersey, from 1999 to 2000, respectively. He was the chairman of the Department of Computer Science and Information Engineering at National Dong Hwa University, Taiwan, from 1995 to 1998, a faculty member in the Department of Electrical and Computer Engineering at the University of Arizona from 1988 to 1991, and an engineer at Fairchild Semiconductor and Silvar-Lisco, both in California, from 1982 to 1984. In 1989, he also worked as a summer faculty fellow at the Jet Propulsion Laboratory of California Institute of Technology. His current research interests include dependable systems and networks, software reliability engineering, mobile computing, and reliable sensor networks. Professor Kuo is an IEEE fellow. He has published more than 240 papers in journals and conferences. He received the distinguished research award between 1997 and 2005, consecutively, from the National Science Council in Taiwan and is now a research fellow there. He was also a recipient of the Best Paper Award in the 1996 International Symposium on Software Reliability Engineering, the Best Paper Award in the simulation and test category at the 1986 IEEE/ACM Design Automation Conference (DAC), the US National Science Foundation's Research Initiation Award in 1989, and the IEEE/ACM Design Automation Scholarship in 1990 and 1991.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.