# Computing System Failure Frequencies and Reliability Importance Measures Using OBDD

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

**Abstract**—The recent literature 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 presents algorithms based on OBDD to compute system failure frequencies and reliability importance measures. Methods are presented to calculate both steady-state and time-specific frequencies of system-failure as well as system-success. The reliability importance measures discussed in this paper include the Birnbaum importance, the Criticality importance, and other indices for the risk evaluation of a system. In addition, we propose an efficient approach based on OBDD to evaluate the reliability of a nonrepairable system and the availability of a repairable system with imperfect fault-coverage mechanisms. The powerful capability of OBDD for reliability evaluation is fully exploited in this paper. Further, we extend all of the proposed algorithms in this paper to analyze systems with imperfect fault-coverage.

**Index Terms**—Failure frequency, reliability importance measure, BDD, imperfect coverage, system availability, fault tolerance.

## 1 Introduction

The reliability analysis of a system includes not only the reliability and availability calculations, but also the calculations of other indices related to reliability. Some of the important indices include the Mean Time To Failure (MTTF), the Mean Time Between Failures (MTBF), the failure frequency, and the reliability importance measures. Recent papers [1], [2], [3], [4], [5], [6], [7] 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 [8] and the sum of disjoint products (SDP) method [9], [10], [11]. However, the advantages of OBDD have not been explored to compute other reliability indices such as system failure frequency and reliability importance measures.

Many researchers have evaluated the system availability (reliability) using various methods, such as the I-E method and the SDP method. However, recent literature on the reliability or availability evaluation of a system with *s*-independent components showed that OBDD is efficient in terms of computational time and accuracy. Although there has been considerable interest in methods on calculating the system failure frequencies [4], [8], [11], [12], [13], [14], there is no convenient method to compute the frequencies using OBDD. Moreover, most of the reported works assume that the process of failure and repair has reached steady state. Techniques to calculating the time-specific and interval frequencies are described in [13], [14], [15]; however, they are limited to availability expressions obtained using the I-E method and the SDP method.

In [16], Schneeweiss showed that it is difficult to calculate failure frequencies using OBDD. However, later, in [17], Schneeweiss has shown a way to compute failure frequency using OBDD. This method is similar to the method proposed in [18] for calculating failure frequencies using Shannon decomposition. The method in [17] did not take full advantage of OBDD structure and is limited to the calculation of only the steady-state frequencies. In [19] Sinnamon and Andrews presented an OBDD-based algorithm to calculate system failure frequency using Birnubaum importance measures of system components. This method does not take full advantage of the relationships between various reliability characteristics; therefore, it needs to traverse the OBDD $n$ times, where $n$ is the total number of components in the system. In this paper, we propose new algorithms to compute steady-state as well as time-specific failure and success frequencies of a system using OBDD. In addition to the calculation of system failure frequencies, we also propose an efficient algorithm based on OBDD for the evaluation of the reliability importance measures of a system. Moreover, these proposed algorithms could be extended to systems with imperfect fault-coverage.

Further, the reliability analysis of a nonrepairable system or a repairable system with imperfect fault-coverage is discussed in this paper. We propose an approach to incorporating the imperfect coverage model for a repairable system into a combinatorial model. The model uses Markov chains for a repairable system with imperfect fault-coverage and is quite useful. In addition, an OBDD-based algorithm is devised based on this approach. Due to the use of conditional probabilities and Markov chains, this OBDD-based algorithm is very efficient for the availability evaluation of a repairable system with imperfect fault-coverage.

---

- *Y.R. Chang and S.Y. Kuo are with the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan. E-mail: raymond@iner.gov.tw, sykuo@cc.ee.ntu.edu.tw.*
- *S.V. Amari is with Relex Software Corp., 540 Pellis Rd., Greensburg, PA 15601. E-mail: suprasad.amari@relexsoftware.com.*

*Manuscript received 31 Dec. 2002; accepted 1 May 2003.*
*For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 117884.*

0018-9340/04/$17.00 © 2004 IEEE Published by the IEEE Computer Society

In Section 2, we propose several OBDD-based algorithms to calculate the system failure frequencies. These algorithms are efficient under not only steady-state condition but also time-specific condition. Section 3 illustrates how we apply the OBDD-based algorithm to the imperfect fault-coverage model for evaluating the reliability of a nonrepairable system and the availability of a repairable system. In addition, the failure frequencies of imperfect coverage models are discussed in Section 4. Also, an algorithm based on OBDD is proposed to calculate the failure frequency of a system with imperfect fault-coverage. In Section 5, efficient algorithms for the evaluation of reliability importance measures, including the Birnbaum importance, the Criticality importance, and other indices for risk evaluation of the system, are proposed. These efficient algorithms are also extended for systems with imperfect fault-coverage. The last section gives the conclusions and future works.

## 2 Failure Frequency

The frequency of failure ($v$), i.e., the mean number of system failures per unit time, is a key parameter in reliability and risk analysis. In general, $v$ can be obtained from the system availability ($A$) or unavailability ($U$) expression as well as the failure and repair rates of components. Schneeweiss [20] indicated that the failure frequency is more important than the system availability. Indeed, the expected system cost depends upon not only the mean down-time (or availability) but also the number of failures during a specific time interval since each repair incurs a certain cost, irrespective of the down-time. Other performance measures of a system, such as the mean up-time and the mean down-time, can be determined easily from the steady-state availability and the steady-state system failure frequency [4], [8].

$$\begin{aligned} \mathrm{MTBF} &= 1/v \\ \mathrm{MTTF} &= A/v \\ \mathrm{MTTR} &= U/v. \end{aligned}$$

When a system is in steady-state condition, the expected number of failures is equivalent to the expected number of successes (from failed states). Therefore, under steady-state conditions, the system failure frequency is equivalent to the system success frequency. Further, MTTF is the expected up-time during a failure and repair cycle. Hence, the expected up-time in a unit time is MTTF/MTBF = $A$. Therefore, for a time interval $T$, we have

$$\begin{aligned} \text{Expected number of failures during } T &= \mathrm{ENF} = vT \\ \text{Expected up-time during } T &= \mathrm{EUT} = AT \\ \text{Expected down-time during } T &= \mathrm{EDT} = UT. \end{aligned}$$

For short duration missions, the time-specific (time-dependent) availability $A(t)$ and the failure frequency $v(t)$ are important. In this case, the system failure frequency ($v_f$) is not equivalent to the system success frequency ($v_s$). Therefore, $v_f$ and $v_s$ should be calculated separately to find other importance measures. Under a transient condition, we have

$$\text{Expected number of successes during } [0,T] = \int_0^T v_s(t)dt$$

$$\text{Expected number of failures during } [0,T] = \int_0^T v_f(t)dt$$

$$\text{Expected up-time during } [0,T] = \mathrm{EUT} = \int_0^T A(t)dt$$

$$\text{Expected down-time during } [0,T] = \mathrm{EDT} = \int_0^T U(t)dt.$$

Therefore, in order to compute the reliability indices of computer systems or networks for reliability analysis, efficient algorithms are needed for evaluating both the system failure frequency and the system availability. In this paper, we propose new and fast algorithms to compute the steady-state as well as time-specific failure and success frequencies of a system using OBDD.

### 2.1 Steady-State Failure Frequency

#### 2.1.1 Assumptions

1. The system is composed of $n$ $s$-independent components. Its structure function (describing redundancy) is $s$-coherent [21], [22].
2. The system and its components have two states: up (working) and down (failed).
3. The failure and repair rates of the components are constant.
4. All components are repairable. Repaired components are as good as new.
5. The system is in steady-state.

#### 2.1.2 Calculating Probability from OBDD

OBDD [23] is based on a disjoint decomposition of Boolean function called the *Shannon expansion*. Given a Boolean function $f(x_1, \cdots, x_n)$, then, for any $i \in \{1, \cdots, n\}$; $\bar{x}_i \equiv \neg x_i = 1 - x_i$:

$$f = x_i \cdot f_{x_i=1} + \bar{x}_i \cdot f_{x_i=0}. \quad (1)$$

In order to express Shannon decomposition concisely, the if-then-else (*ite*) format [24], [25] is defined as:

$$f = ite(x_i, f_{x_i=1}, f_{x_i=0}).$$

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:

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

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

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

where $\diamondsuit$ represents a logic operation such as AND and OR. Fig. 1 illustrates the construction and manipulation steps of

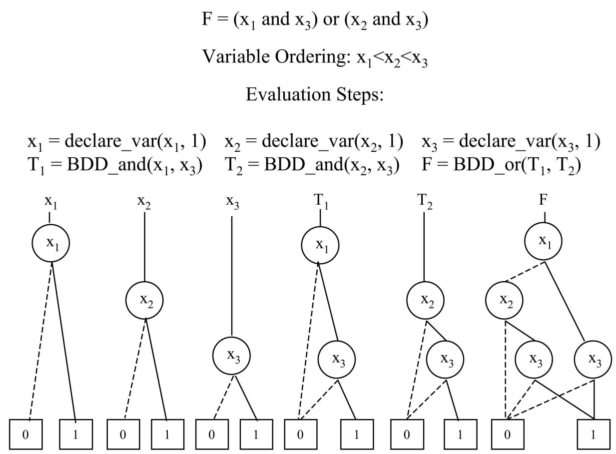


Fig. 1. The OBDD generated from a Boolean equation.

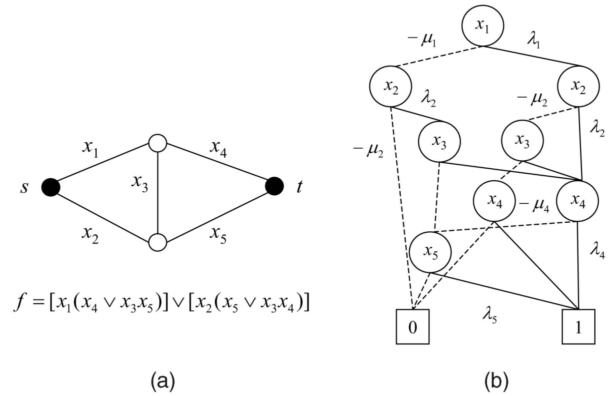


Fig. 2. (a) A bridge nework with failure rate $\lambda_i$ and repair rate $\mu_i$ for component $x_i$ at steady state. (b) The OBDD of (a).

a Boolean function. For more details on using the operations of OBDD, please refer to [23].

A useful property of OBDD is that all the paths from the root to the leaves are mutually disjoint. If $f$ represents the system availability expression, based on the property of the disjoint decomposition of OBDD, the reliability (or availability) of the system can be recursively evaluated by (1).

$$\Pr\{f\} = \Pr\{x_i\} \cdot \Pr\{f_{x_i=1}\} + \Pr\{\bar{x}_i\} \cdot \Pr\{f_{x_i=0}\},$$

where $\Pr\{\cdot\}$ means $\Pr\{\cdot = 1\}$ for simplification. For example, if $\Pr\{x_i\}$ is the availability $A_i$ of component $i$ and $U_i$ is the unavailability of component $i$, then

$$A = \Pr\{f\} = A_iA_{x_i=1} + U_iA_{x_i=0} = A_iA_{x_i=1} + (1 - A_i)A_{x_i=0}, \quad (2)$$

where $A_{x_i=1}(A_{x_i=0})$ is equal to $\Pr\{f_{x_i=1}\}(\Pr\{f_{x_i=0}\})$. Similarly, when the system unavailability expression is used, the unavailability of a system can be calculated as:

$$U = \Pr\{g\} = U_iU_{y_i=1} + A_iU_{y_i=0}, \quad (3)$$

where $y_i = 1$ means component $i$ has failed, $U_{y_i=1}(U_{y_i=0})$ is equal to $\Pr\{g_{y_i=1}\}(\Pr\{g_{y_i=0}\})$, $g$ is the dual of $f$, i.e.,

$$\begin{aligned} f(x_1, x_2, \cdots, x_n) &\equiv 1 - g(1 - x_1, 1 - x_2, \cdots, 1 - x_n) \\ &\equiv 1 - g(y_1, y_2, \cdots, y_n). \end{aligned}$$

### 2.1.3 Calculating Failure Frequency Using OBDD (Method 1)

In this section, we describe an algorithm based on OBDD to calculate the frequency of system failure at steady state. From rule I in [11], $v$ can be derived from the system availability expression $f$, which is expressed in the sum of disjoint product terms $(x_ix_j \cdots \bar{x}_k\bar{x}_l \cdots)$, by multiplying the probability $(A_iA_j \cdots U_kU_l \cdots)$ of every product term $(x_ix_j \cdots \bar{x}_k\bar{x}_l \cdots)$ by $(\lambda_i + \lambda_j + \cdots - \mu_k - \mu_l - \cdots)$, where $\lambda_i(\mu_i)$ is the failure (repair) rate of component $i$. Therefore, from (2) the system failure frequency becomes

$$v = A_i \cdot A_{x_i=1} \cdot [\lambda_i + \lambda_{x_i=1}] + U_i \cdot A_{x_i=0} \cdot [-\mu_i + \lambda_{x_i=0}], \quad (4)$$

where $\lambda_{x_i=1}(\lambda_{x_i=0})$ is the effective failure rate of subterm $f_{x_i=1}(f_{x_i=0})$. Similarly, if the system unavailability expression is used, from rule II in [11], $v$ can be obtained from the system unavailability expression $g$ by multiplying the probability $(A_iA_j \cdots U_kU_l \cdots)$ of every product term $(x_ix_j \cdots \bar{x}_k\bar{x}_l \cdots)$ by $(-\lambda_i - \lambda_j - \cdots + \mu_k + \mu_l + \cdots)$. Therefore, from (3), the system failure frequency is obtained

$$v = U_i \cdot U_{y_i=1} \cdot [\mu_i + \mu_{y_i=1}] + A_i \cdot U_{y_i=0} \cdot [-\lambda_i + \mu_{y_i=0}],$$

where $\mu_{y_i=1}(\mu_{y_i=0})$ is the effective repair rate of subterm $g_{y_i=1}(g_{y_i=0})$.

If the OBDD representing a system has been constructed as shown in Fig. 2, based on the property of disjoint decomposition of OBDD, every path from the root to the leaf node "1" means a disjoint product term of $f$ (i.e., $x_ix_j \cdots \bar{x}_k\bar{x}_l \cdots$). Therefore, we get $(\lambda_i + \lambda_j + \cdots - \mu_k - \mu_l - \cdots)$ by summing up the rates of nodes along the path. For instance, in Fig. 2b, we get the path rate $(\lambda_1 + \lambda_2 + \lambda_4)$ and $(\lambda_1 + \lambda_2 - \mu_4 + \lambda_5)$ from the paths of $\{x_1, x_2, x_4, \mathrm{BDD\_one}\}$ and $\{x_1, x_2, x_4, x_5, \mathrm{BDD\_one}\}$, respectively. Therefore, we modify the general algorithm of computing the probability of OBDD by recording on each node the path rate from the root to it and multiplying the probability $(A_iA_j \cdots U_kU_l \cdots)$ the recorded path rate $(\lambda_i + \lambda_j + \cdots - \mu_k - \mu_l - \cdots)$ when the calculation reaches the leaf node "1." The OBDD-based recursive algorithm for calculating the system failure frequency from the availability expression of the system is shown in Fig. 3. If the calculation is based on unavailability, then the sign of rate should be flipped.

**Example 1.** For a terminal-pair network system, Kuo et al. [3] proposed an efficient approach to determine the terminal-pair (from source node $s$ to target node $t$) reliability based on edge expansion diagrams using OBDD. The main idea, which avoids redundant calculations and makes their approach much more efficient than other methods, is that the OBDD of a given network is automatically constructed with the convergence of isomorphic subproblems during traversing the network from source to sink. Also a good variable-ordering is obtained by breadth-first traversing the network from source to sink in [3]. Therefore, the system reliability is efficiently derived from the OBDD.

```
Procedure double sys_failfreq (BDDnode x_i, double path_rate) {
    double s_rate, f_rate, result;
    if (x_i = BDD_one) then return (path_rate);
    if (x_i = BDD_zero) then return (0);

    s_rate = path_rate + λ_i;
    f_rate = path_rate – μ_i;

    result = A_i * sys_failfreq (sub_node_true(x_i), s_rate)
           + U_i * sys_failfreq (sub_node_false(x_i), f_rate);

    return(result);
}
```

Fig. 3. OBDD-based algorithm using accumulative path rate to calculate the system failure frequency at steady state.

Considering a bridge network as shown in Fig. 2a, the path function of this terminal-pair network system is

$$f = [x_1(x_4 \vee x_3x_5)] \vee [x_2(x_5 \vee x_3x_4)].$$

Fig. 2b shows the OBDD of this network system constructed by [3] during traversing the network from source to sink. The derivation is as follows:

By (1), with $i = 1$:

$$f = x_1[(x_4 \vee x_3x_5) \vee x_2(x_5 \vee x_3x_4)] + \bar{x}_1[x_2(x_5 \vee x_3x_4)].$$

By (1) with $i = 2$ (for a common sub-OBDD):

$$\begin{aligned} f &= x_1[x_2(x_4 \vee x_3x_5 \vee x_5 \vee x_3x_4) + \bar{x}_2(x_4 \vee x_3x_5)] \\ &\quad + \bar{x}_1[x_2(x_5 \vee x_3x_4) + \bar{x}_2(0)]. \end{aligned}$$

Repeat until $i = 6$ and we get

$$\begin{aligned} f &= x_1[x_2(x_4 + \bar{x}_4x_5) + \bar{x}_2(x_3(x_4 + \bar{x}_4x_5) + \bar{x}_3x_4)] \\ &\quad + \bar{x}_1[x_2(x_3(x_4 + \bar{x}_4x_5) + \bar{x}_3x_5)]. \end{aligned}$$

Then, the system availability can be obtained by (2), i.e., a set of mutually node disjoint paths from the root to the leaf node "1" in OBDD.

$$\begin{aligned} A &= A_1 \cdot A_2 \cdot A_4 + A_1 \cdot A_2 \cdot U_4 \cdot A_5 + A_1 \cdot U_2 \cdot A_3 \cdot A_4 \\ &\quad + A_1 \cdot U_2 \cdot A_3 \cdot U_4 \cdot A_5 + A_1 \cdot U_2 \cdot U_3 \cdot A_4 + U_1 \cdot A_2 \cdot A_3 \cdot A_4 \\ &\quad + U_1 \cdot A_2 \cdot A_3 \cdot U_4 \cdot A_5 + U_1 \cdot A_2 \cdot U_3 \cdot A_5. \end{aligned}$$

From (4), the system failure frequency can also be obtained as follows:

$$\begin{aligned} v &= A_1 \cdot A_2 \cdot A_4 \cdot (\lambda_1 + \lambda_2 + \lambda_4) \\ &\quad + A_1 \cdot A_2 \cdot U_4 \cdot A_5 \cdot (\lambda_1 + \lambda_2 - \mu_4 + \lambda_5) \\ &\quad + A_1 \cdot U_2 \cdot A_3 \cdot A_4 \cdot (\lambda_1 - \mu_2 + \lambda_3 + \lambda_4) \\ &\quad + A_1 \cdot U_2 \cdot A_3 \cdot U_4 \cdot A_5 \cdot (\lambda_1 - \mu_2 + \lambda_3 - \mu_4 + \lambda_5) \\ &\quad + A_1 \cdot U_2 \cdot U_3 \cdot A_4 \cdot (\lambda_1 - \mu_2 - \mu_3 + \lambda_4) \\ &\quad + U_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot (-\mu_1 + \lambda_2 + \lambda_3 + \lambda_4) \\ &\quad + U_1 \cdot A_2 \cdot A_3 \cdot U_4 \cdot A_5 \cdot (-\mu_1 + \lambda_2 + \lambda_3 - \mu_4 + \lambda_5) \\ &\quad + U_1 \cdot A_2 \cdot U_3 \cdot A_5 \cdot (-\mu_1 + \lambda_2 - \mu_3 + \lambda_5). \end{aligned} \tag{5}$$

It should be noted that, in order to get the system failure frequency, the previous methods need to find the disjoint terms of the availability expression from the minimal path/cut sets of a network system or assume they are given. In our method, the OBDD is based on the set of disjoint decomposed functions and automatically constructed during traversing the network [3]. Combining our method with [3], it will be very efficient for calculating the failure frequency of a network system.

### 2.1.4 Calculating Failure Frequency Using OBDD (Method 2)

In the above section, once the OBDD is obtained, the time complexity of that algorithm depends on the number of paths in the OBDD. This section describes another algorithm whose time complexity is proportional to the number of nodes in the OBDD.

**Theorem 1.**

$$v = v_i \cdot [A_{x_i=1} - A_{x_i=0}] + A_i \cdot v_{x_i=1} + U_i \cdot v_{x_i=0}, \tag{6}$$

*where*

$$\begin{aligned} v_i &= A_i \cdot \lambda_i \\ v_{x_i=1} &= A_{x_i=1} \cdot \lambda_{x_i=1} \\ v_{x_i=0} &= A_{x_i=0} \cdot \lambda_{x_i=0}. \end{aligned}$$

**Proof.** The proof is based on mathematical induction. From [4], [26], we have

$$v = \sum_{i=1}^{n} v_i \frac{\partial A}{\partial A_i}. \tag{7}$$

From (2), since $A_{x_i=1}$ and $A_{x_i=0}$ do not depend on $A_i$, we have

$$\frac{\partial A}{\partial A_i} = A_{x_i=1} - A_{x_i=0}. \tag{8}$$

Similarly, if $j \neq i$, we have, by (2),

$$\frac{\partial A}{\partial A_j} = A_i \cdot \frac{\partial A_{x_i=1}}{\partial A_j} + U_i \cdot \frac{\partial A_{x_i=0}}{\partial A_j}.$$

Therefore, for any $1 \leq i \leq n$,

$$\begin{aligned} v &= v_i \cdot [A_{x_i=1} - A_{x_i=0}] + \sum_{j=1, j\neq i}^{n} v_j \cdot \left[A_i \frac{\partial A_{x_i=1}}{\partial A_j} + U_i \frac{\partial A_{x_i=0}}{\partial A_j}\right] \\ &= v_i \cdot [A_{x_i=1} - A_{x_i=0}] + A_i \cdot \sum_{j=1, j\neq i}^{n} v_j \frac{\partial A_{x_i=1}}{\partial A_j} + \\ &\quad U_i \cdot \sum_{j=0, j\neq i}^{n} v_j \frac{\partial A_{x_i=0}}{\partial A_j}. \end{aligned}$$

From the definition of failure frequency (see (7)), we know that $\sum_{j=1,j\neq i}^{n} v_j \times \frac{\partial A_{x_i=1}}{\partial A_j}$ is the failure frequency of the system when $x_i = 1$. Similarly, $\sum_{j=1,j\neq i}^{n} v_j \times \frac{\partial A_{x_i=0}}{\partial A_j}$ is the failure frequency of the system when $x_i = 0$. Therefore, (6) holds. Equation (6) is equivalent to (4) if we use the relation $v_i = A_i\lambda_i = U_i\mu_i$ under steady-state condition. □

Equations (2) and (6) form the basic recursion to compute the failure frequency. We can calculate the frequency at each node by first finding the corresponding availability at

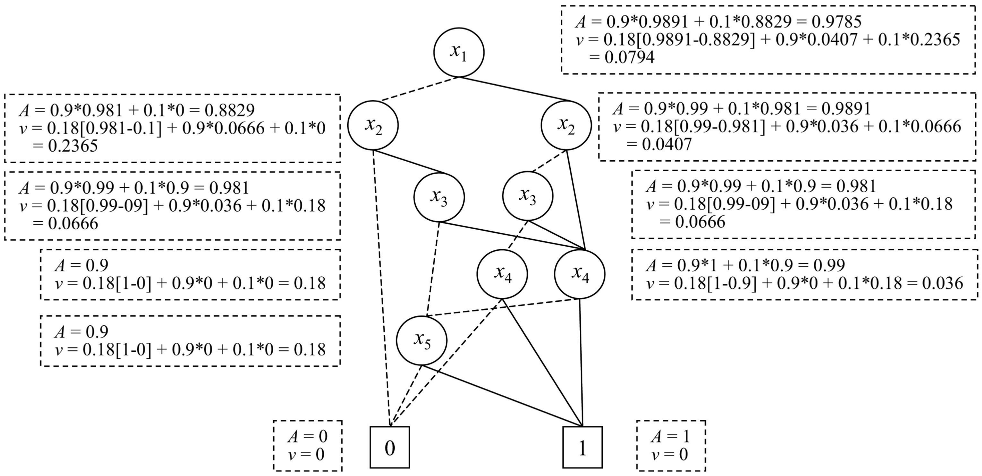


Fig. 4. OBDD of Example 1 with availability = 0.9 and failure rate = 0.2 in each component at steady state.

that node from (2) and then calculating $v$ at that node using (6). These two steps can be applied recursively.

If the OBDD calculations are based on system unavailability, the calculation of failure frequency can also be based on:

**Theorem 2.** $v = v_i \cdot [U_{y_i=1} - U_{y_i=0}] + U_i \cdot v_{y_i=1} + A_i \cdot v_{y_i=0}$, *where*

$$v_i = U_i \cdot \mu_i$$
$$v_{y_i=1} = U_{y_i=1} \cdot \mu_{y_i=1}$$
$$v_{y_i=0} = U_{y_i=0} \cdot \mu_{y_i=0}.$$

**Proof.** The proof is similar to the proof of Theorem 1. □

Applying method 2 to Example 1, the calculation of the system failure frequency is as shown in Fig. 4. The result is the same as that derived by (5). Fig. 5 illustrates the proposed algorithm. Once the OBDD is obtained, the time complexity of the proposed algorithm is linearly proportional to the number of nodes. It should be noted that a hash table is used in this algorithm to avoid repeated calculations in every node of the OBDD. If we get a hit in the hash table, we don't need to recalculate the information of this node. We can retrieve it from the hash table. The hash table used here only records the availability and the failure frequency of each node. An ordinary table could be used instead of a hash table if only a few data need to be stored. However, in [3], the network topology of each node during traversing a path to construct the OBDD is recorded in a hash table. A proper hash table can reduce the comparison time of network topologies. Therefore, our method can be combined with [3] to compute the reliability indices of a network system. This scheme saves significant computational time.

## 2.2 Time-Specific Failure/Success Frequency

### 2.2.1 Assumptions

1. See assumptions in Section 2.1.1.
2. The failure and repair rates of the components are constant.[1]
3. Repaired components are as good as new.[2]

In most of the literature [4], [20], the time-specific frequencies are calculated using the following basic equations:

$$v_f(t) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} E[X_s(t)\bar{X}_s(t + \Delta t)]$$
$$v_s(t) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} E[\bar{X}_s(t)X_s(t + \Delta t)],$$

where $v_f(t)/v_s(t)$ are the time-specific failure/success frequency of a system, i.e., "mean number of system failures/failure-recoveries" per time-unit, $X_s(t)$ is a random variable; $X_s(t) = 1$ means the system is working and $\bar{X}_s(t) = \neg X_s(t)$. $\mathrm{E}(\cdot)$ is the expected value of a random variable.

Under the steady-state condition, we have $v_i = A_i\lambda_i = U_i\mu_i$ and $\mu_i$ is applied directly into (4) to get the system failure frequency. However, under transient conditions (time-specific conditions), the frequency of system failure is not the same as the frequency of system success [13], i.e.,

$$v_f(t) \neq v_s(t) \Leftrightarrow A(t) \cdot \lambda \neq U(t) \cdot \mu.$$

We cannot just apply $\mu_i$ into (4). Therefore, we need to modify our algorithm under time-specific condition.

### 2.2.2 Method 1

From [13], [14], the time-specific system failure and success frequencies are respectively given by

$$v_f(t) = \sum_{i=1}^{n} v_{if}(t)$$
$$v_s(t) = \sum_{i=1}^{n} v_{is}(t), \tag{9}$$

where $v_{if}(t)/v_{is}(t)$ is the contribution of component $i$ to the system time-specific frequency of failure/success. Further, since component $i$ is $s$-independent of the rest of the system

1. The proposed algorithm can also be applied to the case of global time-dependent failure and repair rates. In this case, the individual component availabilities should be calculated using appropriate techniques.

2. This assumption would not be applicable if global time varying failure and repair rates are considered.

```
struct nodeInfo {
    double A;  // availability
    double v;  // failure frequency
};
Procedure nodeInfo subsys_fail_freq (BDDnode x_i) {
    nodeInfo n_true, n_false, result;
    if (x_i = BDD_one) then {
        result.A = 1;
        result.v = 0;
        return (result);
    }
    if (x_i = BDD_zero) then {
        result.A = 0;
        result.v = 0;
        return (result);
    }
    if (result=get_computed_table(x_i) is hit) then return (result);

    n_true = subsys_fail_freq (sub_node_true(x_i));
    n_false = subsys_fail_freq (sub_node_false(x_i));

    result.A = A_i * n_true.A + U_i * n_false.A;
    result.v = A_i * λ_i * [n_true.A – n_false.A]
             + A_i * n_true.v + U_i * n_false.v;

    insert_computed_table(x_i, result);
    return (result);
}
```

Fig. 5. Algorithm for calculating availability and failure frequency.

and after changing the notation appropriately, the contribution frequencies [14] are

$$\begin{aligned} v_{if}(t) &= [A_{x_i=1}(t) - A_{x_i=0}(t)] \cdot A_i(t) \cdot \lambda_i \\ &= [A_{y_i=0}(t) - A_{y_i=1}(t)] \cdot A_i(t) \cdot \lambda_i \\ &= [U_{x_i=0}(t) - U_{x_i=1}(t)] \cdot A_i(t) \cdot \lambda_i \\ &= [U_{y_i=1}(t) - U_{y_i=0}(t)] \cdot A_i(t) \cdot \lambda_i \end{aligned} \tag{10}$$

and

$$\begin{aligned} v_{is}(t) &= [U_{y_i=1}(t) - U_{y_i=0}(t)] \cdot U_i(t) \cdot \mu_i \\ &= [U_{x_i=0}(t) - U_{x_i=1}(t)] \cdot U_i(t) \cdot \mu_i \\ &= [A_{y_i=0}(t) - A_{y_i=1}(t)] \cdot U_i(t) \cdot \mu_i \\ &= [A_{x_i=1}(t) - A_{x_i=0}(t)] \cdot U_i(t) \cdot \mu_i, \end{aligned} \tag{11}$$

where $x_i = \neg y_i$. By combining (9), (10), and (11) and applying rule A and rule B in [14], we get

$$\begin{aligned} v_f(t) &= \sum_{j \in I} g_j(t) \left[ \sum_{i \in Z_j} \lambda_i - \sum_{i \in \bar{Z}_j} (\lambda_i A_i(t)/U_i(t)) \right] \\ v_s(t) &= \sum_{j \in I} g_j(t) \left[ \sum_{i \in Z_j} (\mu_i U_i(t)/A_i(t)) - \sum_{i \in \bar{Z}_j} \mu_i \right], \end{aligned} \tag{12}$$

where $I$ is the set of subterms of expression of $A(t)$, $g_j(t)$ is the subterm of index $j$ in the expression of $A(t)$, $Z_j$ and $\bar{Z}_j$ are index subsets such that members of $\{A_i(t)\}_{i \in Z_j}$ and $\{U_i(t)\}_{i \in \bar{Z}_j}$ comprise $g_j(t)$. These are basic equations for the time-specific case. In fact, for all subsystems, if we define

$$A(t) \cdot \lambda' = U(t) \cdot \mu \Rightarrow \lambda' = \frac{U(t) \cdot \mu}{A(t)}$$
$$U(t) \cdot \mu' = A(t) \cdot \lambda \Rightarrow \mu' = \frac{A(t) \cdot \lambda}{U(t)}$$

and make some modifications from (4), then we get

$$\begin{aligned} v_f(t) = A_i(t) \cdot A_{x_i=1}(t) \cdot [\lambda_i + \lambda_{x_i=1}] + U_i(t) \cdot A_{x_i=0}(t) \cdot \\ [-\mu'_i + \lambda_{x_i=0}]. \end{aligned} \tag{13}$$

Equation (13) is equivalent to (12). Similarly, we can get the following equations:

$$\begin{aligned} v_f(t) &= U_i(t) \cdot U_{y_i=1}(t) \cdot [\mu'_i + \mu'_{y_i=1}] + A_i(t) \cdot U_{y_i=0}(t) \cdot \\ &\quad [-\lambda_i + \mu'_{y_i=0}] \\ v_s(t) &= A_i(t) \cdot A_{x_i=1}(t) \cdot [\lambda'_i + \lambda'_{x_i=1}] + U_i(t) \cdot A_{x_i=0}(t) \cdot \\ &\quad [-\mu_i + \lambda'_{x_i=0}] \\ v_s(t) &= U_i(t) \cdot U_{y_i=1}(t) \cdot [\mu_i + \mu_{y_i=1}] + A_i(t) \cdot U_{y_i=0}(t) \cdot \\ &\quad [-\lambda'_i + \mu_{y_i=0}]. \end{aligned}$$

Therefore, the modified algorithm is similar to the steady-state algorithm. The only difference is that we should use $\mu'_i$ instead of $\mu_i$ to compute the time-specific frequency of system failure based on the availability expression, where $\mu'_i = \lambda_i A_i(t)/U_i(t)$. To simplify the representation, we use $A_i$ ($U_i$) to represent $A_i(t)$ ($U_i(t)$), i.e., the time-specific availability (unavailability) of component $i$, and (5) becomes

$$\begin{aligned} v_f(t); &= A_1 \cdot A_2 \cdot A_4 \cdot (\lambda_1 + \lambda_2 + \lambda_4) \\ &+ A_1 \cdot A_2 \cdot U_4 \cdot A_5 \cdot (\lambda_1 + \lambda_2 - \lambda_4 A_4/U_4 + \lambda_5) \\ &+ A_1 \cdot U_2 \cdot A_3 \cdot A_4 \cdot (\lambda_1 - \lambda_2 A_2/U_2 + \lambda_3 + \lambda_4) \\ &+ A_1 \cdot U_2 \cdot A_3 \cdot U_4 \cdot A_5 \cdot (\lambda_1 - \lambda_2 A_2/U_2 + \lambda_3 - \lambda_4 A_4/U_4 + \lambda_5) \\ &+ A_1 \cdot U_2 \cdot U_3 \cdot A_4 \cdot (\lambda_1 - \lambda_2 A_2/U_2 - \lambda_3 A_3/U_3 + \lambda_4) \\ &+ U_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot (-\lambda_1 A_1/U_1 + \lambda_2 + \lambda_3 + \lambda_4) \\ &+ U_1 \cdot A_2 \cdot A_3 \cdot U_4 \cdot A_5 \cdot \\ &\quad (-\lambda_1 A_1/U_1 + \lambda_2 + \lambda_3 - \lambda_4 A_4/U_4 + \lambda_5) \\ &+ U_1 \cdot A_2 \cdot U_3 \cdot A_5 \cdot (-\lambda_1 A_1/U_1 + \lambda_2 - \lambda_3 A_3/U_3 + \lambda_5). \end{aligned}$$

If the calculations are based on unreliability, then signs should be flipped. The modified algorithm for the calculation of the time-specific failure frequency is shown in Fig. 6. Also, for the computation of time-specific frequency of system success ($v_s(t)$), we should use $\lambda'_i$ instead of $\lambda_i$, where $\lambda'_i = \mu_i U_i(t)/A_i(t)$.

### 2.2.3 Method 2

In this section, we describe another method with time complexity proportional to the number of nodes in the OBDD. Let $v_{fi}(t)/v_{si}(t)$ be the time-specific frequency of failure/success of component $i$ and, by (8), we have

$$\begin{aligned} v_{fi}(t) &= A_i(t) \cdot \lambda_i \\ v_{si}(t) &= U_i(t) \cdot \mu_i \\ \frac{\partial A(t)}{\partial A_i(t)} &= [A_{x_i=1}(t) - A_{x_i=0}(t)] \\ \frac{\partial U(t)}{\partial U_i(t)} &= [U_{y_i=1}(t) - U_{y_i=0}(t)]. \end{aligned}$$

```
Procedure double sys_failfreq (BDDnode x_i, double path_rate) {
    double s_rate, f_rate result;
    if (x_i = BDD_one) then return (path_rate);
    if (x_i = BDD_zero) then return (0);

    s_rate = path_rate + λ_i;
    f_rate = path_rate – λ_i *A_i / U_i;

    result = A_i * sys_failfreq (sub_node_true(x_i), s_rate)
           + U_i * sys_failfreq (sub_node_false(x_i), f_rate);

    return (result);
}
```

Fig. 6. The modified algorithm for calculating the time-specific frequency of system failure.

Therefore, (9) can be expressed as below:

$$\begin{aligned} v_f(t) &= \sum_{i=1}^{n} \frac{\partial A(t)}{\partial A_i(t)} v_{fi}(t) = \sum_{i=1}^{n} \frac{\partial U(t)}{\partial U_i(t)} v_{fi}(t) \\ v_s(t) &= \sum_{i=1}^{n} \frac{\partial A(t)}{\partial A_i(t)} v_{si}(t) = \sum_{i=1}^{n} \frac{\partial U(t)}{\partial U_i(t)} v_{si}(t). \end{aligned} \tag{14}$$

Equation (14) is similar to (7). This method for calculating the time-specific frequencies is very much similar to the case under steady-state condition, except we should use $v_{fi}(t)$ instead of $v_i(t)$ to calculate $v_f(t)$ and use $v_{si}(t)$ instead of $v_i(t)$ to calculate $v_s(t)$. Therefore, we get:

$$\begin{aligned} v_f(t) &= v_{fi}(t) \cdot [A_{x_i=1}(t) - A_{x_i=0}(t)] + A_i(t) \cdot v_{f_{|x_i=1}}(t) \\ &\quad + U_i(t) \cdot v_{f_{|x_i=0}}(t) \\ &= v_{fi}(t) \cdot [U_{y_i=1}(t) - U_{y_i=0}(t)] + U_i(t) \cdot v_{f_{|y_i=1}}(t) \\ &\quad + A_i(t) \cdot v_{f_{|y_i=0}}(t), \end{aligned} \tag{15}$$

$$\begin{aligned} v_s(t) &= v_{si}(t) \cdot [A_{x_i=1}(t) - A_{x_i=0}(t)] + A_i(t) \cdot v_{s_{|x_i=1}}(t) \\ &\quad + U_i(t) \cdot v_{s_{|x_i=0}}(t) \\ &= v_{si}(t) \cdot [U_{y_i=1}(t) - U_{y_i=0}(t)] + U_i(t) \cdot v_{s_{|y_i=1}}(t) \\ &\quad + A_i(t) \cdot v_{s_{|y_i=0}}(t). \end{aligned} \tag{16}$$

The proof is similar to the proof of (6).

In fact, from (15), we have

$$\begin{aligned} v_f(t) &= A_i(t) \cdot \lambda_i \cdot [A_{x_i=1}(t) - A_{x_i=0}(t)] + A_i(t) \cdot A_{x_i=1}(t) \cdot \\ &\quad \lambda_{x_i=1} + U_i(t) \cdot A_{x_i=0}(t) \cdot \lambda_{x_i=0} \\ &= A_i(t) \cdot A_{x_i=1}(t) \cdot [\lambda_i + \lambda_{x_i=1}] \\ &\quad + U_i(t) \cdot A_{x_i=0}(t) \cdot \left[ -\frac{\lambda_i \cdot A_i(t)}{U_i(t)} + \lambda_{x_i=0} \right]. \end{aligned}$$

This equation is equivalent to (13). It should be noted that, by using this method, we don't need to find the modified instantaneous rates $\mu_i'$ or $\lambda_i'$ as in Method 1. Therefore, we should use (15) and (16) to compute both $v_f(t)$ and $v_s(t)$ either from the availability or the unavailability based on OBDD. The time complexity of this method is also proportional to the number of nodes in the OBDD.

# 3 Availability under Imperfect Coverage Model

Computer systems that are used in life-critical applications, such as flight control, nuclear power plant monitoring, space missions, etc., are designed with sufficient redundancy to be tolerant of errors that may occur. However, if a system cannot detect, locate, and recover from faults and errors in a system, then system failures can result even when there is enough redundancy [26]. Therefore, an accurate analysis must account for not only the complex system structure, but also the system fault and error recovery behavior as well. This helps in determining the optimal level of redundancy. Otherwise, the increase in redundancy may decrease the system reliability (availability) due to imperfect coverage, as shown by Pham [28] for the reliability of Triple Modular Redundant Systems and for more general cases in [29].

*Assumptions*

1. Component failures are *s*-independent.
2. An *s*-coherent combinatorial model (fault-tree or digraph or network) can be used to represent the combinations of covered component faults that lead to system fault.
3. Uncovered component failures cause immediate system failure, even in the presence of adequate redundancy.
4. Fault-occurrence probabilities are given: a) as fixed probabilities (for a given mission time) or b) in terms of a lifetime distribution.
5. Coverage = Pr{system can recover | a fault occurs} is given either as a fixed probability or in terms of a model from which coverage factors are derived.

## 3.1 Coverage Model

Fig. 7 shows the general structure of a fault-coverage model representing a recovery process [21], [30] initiated when a fault occurs. The entry point to the model signifies the occurrence of a failure and the three exits (*R*, *S*, *C*) represent 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 using an appropriate fault-coverage model [30]; for more details, see [22], [26].

For the fault-coverage model, each component is always in one of three states: $x[i]$, $y[i]$, $z[i]$. To determine the system reliability (unreliability), it is required to have $a[i]$, $b[i]$, $c[i]$, which represent the probabilities of component $i$ associated with the exits of the fault-coverage model. Fig. 8 shows the

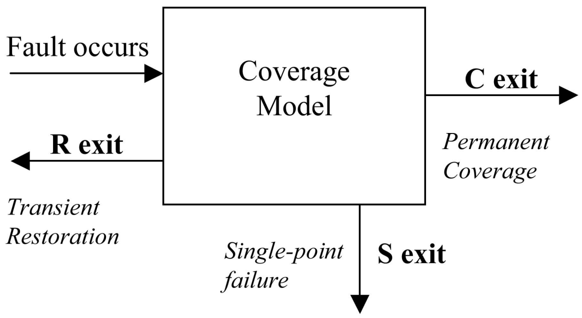


Fig. 7. General structure of a fault-coverage model.

event space (and corresponding probability) representation of a component. Therefore,

$$\begin{aligned} a[i] &= \exp[-(1-r_{i0}) \cdot \lambda_{i0} \cdot t] \\ b[i] &= \frac{c_{i0}}{c_{i0}+s_{i0}} \cdot [1-\exp[-(1-r_{i0}) \cdot \lambda_{i0} \cdot t]] \\ c[i] &= \frac{s_{i0}}{c_{i0}+s_{i0}} \cdot [1-\exp[-(1-r_{i0}) \cdot \lambda_{i0} \cdot t]], \end{aligned} \tag{17}$$

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

$$\begin{aligned} \lambda_i &\equiv (c_{i0}+s_{i0}) \cdot \lambda_{i0} = (1-r_{i0}) \cdot \lambda_{i0} \\ c_i &\equiv c_{i0}(c_{i0}+s_{i0}). \end{aligned} \tag{18}$$

Amari et al. [21] proposed an efficient algorithm, named SEA, to calculate the reliability of a system under the imperfect coverage model (IPCM). SEA separates the fault-coverage modeling of failures into two terms that are multiplied to compute the system reliability. The first term, a simple product, represents the probability that no uncovered fault occurs. The second term comes from a combinatorial model which includes the covered faults that can lead to system failure. This second term can be computed from any common approach (e.g., fault tree, block diagram, digraph) which ignores the fault-coverage concept by slightly altering the component-failure probabilities. The basic idea is shown in the following equation and could be easily proven [21] by using conditional probabilities.

$$\begin{aligned} &\text{System Unavailability} = \Pr\{\text{an uncovered failure}\} \\ &\times \Pr\{\text{system failure} \mid \text{an uncovered failure}\} \\ &+ \Pr\{\text{no uncovered failure}\} \\ &\times \Pr\{\text{system failure} \mid \text{no uncovered failure}\}. \end{aligned}$$

The SEA algorithm [21] focused on nonrepairable systems and works very well. However, we found that the use of conditional probabilities in the SEA algorithm could be extended to repairable systems. Also, the powerful capability of OBDD in computing system reliability could be included in this algorithm.

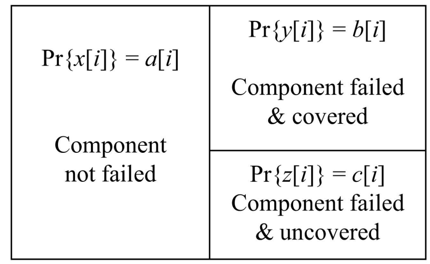


Fig. 8. Event and probability space of component $i$.

### 3.2 Imperfect Fault Coverage Model for Repairable Systems

For a repairable system, the availability (unavailability) is an important index for system performance. Dugan and Trivedi [30] used imperfect-coverage modeling for repairable systems. Akhtar [31] attempted to model the imperfect fault-coverage of a repairable system. Yin et al. [32] considered the components of a system to be repairable and subject to imperfect fault-coverage. They modeled the components using Markov chains. If the components can be repaired, each component will be in any one of its three states at any time. The three states are *good (working) state*, *failed and covered state*, and *failed and uncovered state*. These three states are mutually exclusive in a repairable system. The behavior of a component is shown in Fig. 9, where $\lambda_i$ is the failure rate of component $i$, $\mu_i$ is the repair rate, and $c_i$ is the coverage factor.

Let $S_{ji}(t) = \Pr\{\text{state of component } i \text{ is } j \text{ at time } t\}$, where $j = [1, 2, 3]$ for [good, failed & covered, failed & uncovered], respectively, then

$$\begin{aligned} S_{1i}(t) &= \frac{\mu_i - \alpha_{1i}}{\alpha_{2i} - \alpha_{1i}} e^{-\alpha_{1i} t} + \frac{\mu_i - \alpha_{2i}}{\alpha_{1i} - \alpha_{2i}} e^{-\alpha_{2i} t} \\ S_{2i}(t) &= \frac{\lambda_i c_i}{\alpha_{2i} - \alpha_{1i}} e^{-\alpha_{1i} t} + \frac{\lambda_i c_i}{\alpha_{1i} - \alpha_{2i}} e^{-\alpha_{2i} t} \\ S_{3i}(t) &= 1 - S_{1i}(t) - S_{2i}(t) \\ &= 1 - \frac{\mu_i + \lambda_i c_i - \alpha_{1i}}{\alpha_{2i} - \alpha_{1i}} e^{-\alpha_{1i} t} - \frac{\mu_i + \lambda_i c_i - \alpha_{2i}}{\alpha_{1i} - \alpha_{2i}} e^{-\alpha_{2i} t}, \end{aligned} \tag{19}$$

where

$$\alpha_{1i}, \alpha_{2i} = \frac{\lambda_i + \mu_i \mp \sqrt{(\lambda_i + \mu_i)^2 - 4(1-c_i)\lambda_i \mu_i}}{2}.$$

Therefore, the conditional probabilities (given no uncovered failures) of component $i$ are as follows:

$$\begin{aligned} P_i(t) &= \frac{S_{1i}(t)}{S_{1i}(t) + S_{2i}(t)} \\ Q_i(t) &= 1 - P_i(t) = \frac{S_{2i}(t)}{S_{1i}(t) + S_{2i}(t)}. \end{aligned} \tag{20}$$

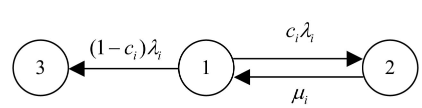


Fig. 9. Markov model of a repairable system subject to IPCM.

```
main() {
    Pu = 1;
    for each component i {
        Pu = Pu * (S1[i]+S2[i]);
    }
    As = Pu * avail(BDDnode root);
    return(As);
}
Procedure double avail(BDDnode xi) {
    double p, q, result;
    p = S1[xi] / (S1[xi] + S2[xi]);
    q = S2[xi] / (S1[xi] + S2[xi]);   // q = 1 – p
    if (node = bdd_one) then return (1);
    if (node = bdd_zero) then return (0);
    if (result = get_computed_table(xi) is hit) then return(result);

    result = p * avail(sub_node_true(xi)) + q * avail(sub_node_false (xi));

    insert_computed_table(xi, result);
    return(result);
}
```

Fig. 10. The SEA algorithm based on OBDD with IPCM in a repairable system.

And, the probability of no uncovered failure of the system, $P_u(t)$ becomes

$$P_u(t) = \prod_{i \in S} (1 - S_{3i}(t)) = \prod_{i \in S} (S_{1i}(t) + S_{2i}(t)), \tag{21}$$

where $S$ is the set of components ($n$ elements).

The SEA algorithm based on OBDD for a repairable system with imperfect coverage is as follows:

1. Find

$$\begin{aligned}\Pr\{\text{no uncovered failure}\} &= P_u(t) \\ &= \prod_{i \in S} (S_{1i}(t) + S_{2i}(t)).\end{aligned}$$

2. Find the conditional probability (given no uncovered failures) of component $i$, $P_i(t)$, $Q_i(t)$, where $P_i(t) = S_{1i}(t)/(S_{1i}(t) + S_{2i}(t))$ and

$$Q_i(t) = 1 - P_i(t) = S_{2i}(t)/(S_{1i}(t) + S_{2i}(t)).$$

3. Using $P_i(t)$, $Q_i(t)$ from Step 2 instead of the availability and unavailability of each component $i$, respectively, find the conditional unavailability ($U_c(t)$) or availability ($A_c(t)$) of a system with perfect coverage by using OBDD.
4. Find the unavailability ($U_s(t)$) for a system with imperfect coverage using

$$\begin{aligned}&\Pr\{\text{an uncovered failure}\} = [1 - P_u(t)] \\ &\Pr\{\text{system failure} \mid \text{an uncovered failure}\} = 1 \\ &\Pr\{\text{no uncovered failure}\} = P_u(t) \\ &\Pr\{\text{system failure} \mid \text{no uncovered failure}\} = U_c(t).\end{aligned}$$

Therefore, we get the following results

$$\begin{aligned}U_s(t) &= [1 - P_u(t)] \cdot 1 + [P_u(t)] \cdot U_c = 1 - P_u(t) \cdot A_c(t) \\ A_s(t) &= 1 - U_s(t) = P_u(t) \cdot A_c(t).\end{aligned} \tag{22}$$

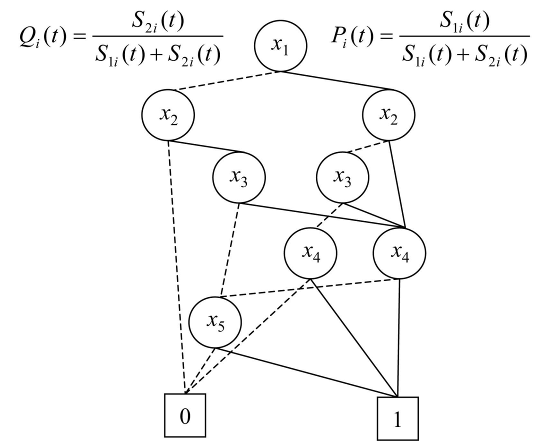


Fig. 11. The availability evaluation of a bridge network subject to imperfect coverage.

For a nonrepairable system, the repair rate $\mu_i$ of component $i$ is zero. Therefore, letting $\mu_i = 0$ in (19), we get the same result, $S_{1i}(t) = a[i]$, $S_{2i}(t) = b[i]$, $S_{3i}(t) = c[i]$ of a nonrepairable system, as (17) and (18). This method is more efficient than [7]. In [7], the method splits one node into two subnodes to represent imperfect coverage. It makes the number of nodes in OBDD become larger and increases the computational complexity. On the other hand, using the conditional probabilities in our method, the computational complexity is the same as that of the method for solving perfect coverage problems. Moreover, our method could also be applied to modular structures using the concept of conditional probabilities. Fig. 10 illustrates the OBDD-based algorithm modified from the SEA algorithm for repairable systems with imperfect coverage.

**Example 2.** Consider a bridge network in Example 1 as shown in Fig. 2a. Assume redundancy techniques are used such that each link has a fault tolerance scheme and is repairable. We can get the system conditional availability $A_c(t)$ by using $P_i(t)$ and $Q_i(t)$ instead of the availability and unavailability of each component $i$ as shown in Fig. 11. Therefore, the availability of this repairable network system subject to imperfect coverage can be easily obtained from (22). With this efficient integration, we don't need to solve the whole problem (in this case, the total number of states is $5^3 = 125$, including the imperfect coverage state) using Markov chain, especially when the network is quite large and complex. Moreover, using the conditional probabilities, this method could be applied to modular structures and the computational complexity of this method is the same as that of the method for solving perfect coverage problems.

### 3.3 Semi-Markov Model

The algorithm proposed above could also be applied to the case where time-to-failure distribution is not exponential. This is a more general case. The behavior of an individual component is shown in Fig. 12.

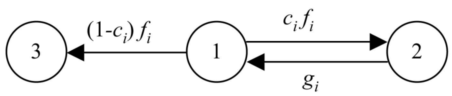


Fig. 12. Semi-Markov model with nonexponential time-to-failure distribution.

*Assumptions*

1. The probability density function (*pdf*), cumulative density function (*cdf*), and survival function (*sf*) of times-to-failure of component $i$ are $f_i(t)$, $F_i(t)$, and $\bar{F}_i(t)$, respectively.
2. The *pdf*, *cdf*, and *sf* of repair times of component $i$ are $g_i(t)$, $G_i(t)$, and $\bar{G}_i(t)$, respectively.
3. The coverage factor of component $i$ is $c_i$.
4. Initially, all components are in good condition.

Using the Semi-Markovian approach, the probabilities of $S_{ji}(t)$ are calculated as follows:

$$\begin{aligned} S_{1i}(t) &= \bar{F}_i(t) + c_i[F_i * G_i * \bar{F}_i(t)] + (c_i)^2\left[F_i^{(2)} * G_i^{(2)} * \bar{F}_i(t)\right] \\ &\quad + \cdots \\ &= \sum_{m=0}^{\infty}(c_i)^m\left[F_i^{(m)} * G_i^{(m)} * \bar{F}_i(t)\right]; F^{(0)} = G^{(0)} \equiv 1 \\ &= \sum_{m=0}^{\infty}(c_i)^m\left[F_i^{(m)} * G_i^{(m)}(t) - F_i^{(m+1)} * G_i^{(m)}(t)\right], \end{aligned} \tag{23}$$

where

$$F * G(t) = \int_0^t F(t-u)dG(u) = \int_0^t G(t-u)dF(u),$$

i.e., the convolution of $F(t)$ and $G(t)$ and $F^{(n)}(t)$ is $n$-fold convolution of $F$ with itself. Similarly,

$$\begin{aligned} S_{2i}(t) &= c_i\left[F_i * \bar{G}_i(t)\right] + (c_i)^2\left[F_i^{(2)} * G_i^{(1)} * \bar{G}_i(t)\right] + \cdots \\ &= \sum_{m=1}^{\infty}(c_i)^m\left[F_i^{(m)} * G_i^{(m-1)} * \bar{G}_i(t)\right] \\ &= \sum_{m=1}^{\infty}(c_i)^m\left[F_i^{(m)} * G_i^{(m-1)}(t) - F_i^{(m)} * G_i^{(m)}(t)\right] \end{aligned} \tag{24}$$

and $S_{3i}(t) = 1 - S_{1i}(t) - S_{2i}(t)$. Solving (23) and (24) is equivalent to solving the convolution and renewal functions [33], [34].

### 3.4 Time-Dependent Coverage Factor

In the above case, it is assumed that the coverage factor is independent of the occurrence time of a fault. However, in some cases, it might depend upon the occurrence time. In other words, occurrence time distributions of covered faults and uncovered faults might be different. Therefore, different distributions for time to occurrence of covered and uncovered faults are considered. Further, if the occurrence rates are constant, the coverage factor becomes constant with respect to time. The behavior of an individual module is shown in Fig. 13.

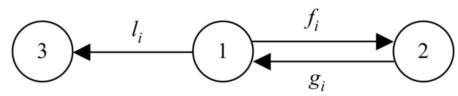


Fig. 13. Semi-Markov model with time-dependent coverage factor.

*Assumptions*

1. The *pdf*, *cdf*, and *sf* of the number of covered faults of component $i$ are $f_i(t)$, $F_i(t)$, and $\bar{F}_i(t)$, respectively.
2. The *pdf*, *cdf*, and *sf* of the number of uncovered faults of component $i$ are $l_i(t)$, $L_i(t)$, and $\bar{L}_i(t)$, respectively.
3. The *pdf*, *cdf*, and *sf* of the number of repairs of component $i$ are $g_i(t)$, $G_i(t)$, and $\bar{G}_i(t)$, respectively.
4. Initially, all components are in good condition.

Using the Semi-Markovian approach, the probabilities of $S_{ji}(t)$ are calculated as follows:

$$\begin{aligned} S_{1i}(t) &= \sum_{m=0}^{\infty}\left[T_i^{(m)} * G_i^{(m)} * N_i(t)\right] \\ S_{2i}(t) &= \sum_{m=0}^{\infty}\left[T_i^{(m+1)} * G_i^{(m)} * \bar{G}_i(t)\right] \\ S_{3i}(t) &= 1 - S_{1i}(t) - S_{2i}(t), \end{aligned}$$

where $T_i(t) = \int_0^t \bar{L}_i(x)dF_i(x)$ and $N_i(t) = \bar{L}_i(t)\bar{F}_i(t)$.

## 4 Failure Frequency under Imperfect Coverage Model

This section describes the calculation of failure frequency under imperfect fault-coverage. An algorithm based on OBDD is proposed in this section. For simplification, we consider the nonrepairable system and use $R$ to represent the time-specific reliability $R(t)$ of a system and $r_i$ the time-specific reliability $r_i(t)$ of component $i$. Then, (17) becomes

$$\begin{aligned} a[i] &= r_i \\ b[i] &= (1 - r_i)c_i \\ c[i] &= (1 - r_i)(1 - c_i). \end{aligned}$$

Since component $i$ is $s$-independent of the rest of the system, from (10), the contribution to failure frequency is

$$v_{if\_IPCM}(t) = [R_{x_i=1}(t) - R_{x_i=0}(t)] \cdot r_i(t) \cdot \lambda_i = \frac{\partial R}{\partial r_i} r_i \lambda_i.$$

Therefore, the failure frequency of the system subject to imperfect coverage is

$$v_{f\_IPCM}(t) = \sum_{i=1}^{n} v_{if\_IPCM}(t) = \sum_{i=1}^{n} \frac{\partial R}{\partial r_i} r_i \lambda_i.$$

As shown in (22) as well as from [21], we can specify the reliability of a system subject to imperfect coverage as follows:

$$R = P_u \cdot R_c, \tag{25}$$

where $P_u$ is the probability of no uncovered failure in the system. $R_c$ is the conditional reliability of the system given that no uncovered failure has occurred in the system and, from (21), we have

$$P_u = \prod_{i=1}^{n} [r_i + (1 - r_i)c_i],$$

where $r_i$ is the reliability of component $i$ and $c_i$ is the coverage factor of component $i$. Let the conditional reliability of component $i$ (given no uncovered failure) as shown in (20) be

$$R_i = \frac{r_i}{r_i + (1 - r_i)c_i}.$$

By (25), we have

$$\frac{\partial R}{\partial r_i} = R_c \frac{\partial P_u}{\partial r_i} + P_u \frac{\partial R_c}{\partial r_i},$$

where

$$\begin{aligned}\frac{\partial P_u}{\partial r_i} &= \prod_{j=1, j\neq i}^{n} \left[r_j + (1 - r_j)c_j\right] \cdot (1 - c_i) \\ &= P_u \frac{1 - c_i}{r_i + (1 - r_i)c_i} = P_u(1 - c_i)\frac{R_i}{r_i}\end{aligned}$$

and, by the chain rule of differentiation,

$$\begin{aligned}\frac{\partial R_c}{\partial r_i} &= \frac{\partial R_c}{\partial R_i} \cdot \frac{\partial R_i}{\partial r_i} \\ \frac{\partial R_i}{\partial r_i} &= \frac{r_i + (1 - r_i)c_i - r_i(1 - c_i)}{(r_i + (1 - r_i)c_i)^2} \\ &= \frac{c_i}{(r_i + (1 - r_i)c_i)^2} = c_i \left(\frac{R_i}{r_i}\right)^2.\end{aligned}$$

Therefore, the failure frequency of the system subject to imperfect coverage is

$$\begin{aligned}v_{f\_IPCM}(t) &= \sum_{i=1}^{n} v_{if\_IPCM}(t) = \sum_{i=1}^{n} \frac{\partial R}{\partial r_i} r_i \lambda_i \\ &= \sum_{i=1}^{n} \left[R_c P_u (1 - c_i)\frac{R_i}{r_i} + P_u \frac{\partial R_c}{\partial R_i} c_i (\frac{R_i}{r_i})^2\right] r_i \lambda_i \\ &= P_u \cdot \left(R_c \sum_{i=1}^{n} w_{1i} + \sum_{i=1}^{n} w_{2i} \frac{\partial R_c}{\partial R_i}\right) \\ &= P_u \cdot \left(v_{f\_IPCM1}(t) + v_{f\_IPCM2}(t)\right),\end{aligned}$$

where

$$\begin{aligned}w_{1i} &= (1 - c_i)\lambda_i R_i \\ w_{2i} &= \frac{c_i}{r_i} \lambda_i R_i^2 \\ v_{f\_IPCM1}(t) &= R_c \sum_{i=1}^{n} w_{1i} \\ v_{f\_IPCM2}(t) &= \sum_{i=1}^{n} w_{2i} \frac{\partial R_c}{\partial R_i}.\end{aligned}$$

In order to obtain $v_{f\_IPCM1}(t)$, we can apply the OBDD method depicted in Section 3 to get $R_c$, i.e., the conditional reliability of a system given that no uncovered failure has occurred, and then $v_{f\_IPCM1}(t)$ can be easily obtained. Moreover, since $v_{f\_IPCM2}(t)$ is similar to $v$ in (7), we can compute frequencies as shown below.

$$\begin{aligned}v_{f\_IPCM2}(t) &= w_{2i} \cdot [R_{c_{|x_i=1}}(t) - R_{c_{|x_i=0}}(t)] \\ &+ R_i(t) \cdot v_{f\_IPCM2_{|x_i=1}}(t) + \tilde{R}_i(t) \cdot v_{f\_IPCM2_{|x_i=0}}(t).\end{aligned}$$

The proof is similar to that for (6). An OBDD-based algorithm similar to that in Fig. 5 can be applied to this equation.

## 5 Reliability Importance Measures

Some components are more critical than others to the functioning of a system as a result of the system's structural arrangement. In addition, a component's failure probability will also have a great influence in assessing its importance with respect to the overall state of a system. Therefore, the purpose of evaluating reliability importance measures (also known as sensitivity analysis) is to obtain information concerning a component's contribution to the system reliability, which can be very useful in system design, failure diagnosis, and system failure probability minimization. For more details on the definitions, please refer to [9].

In [19], two algorithms are proposed to compute the Birnbaum importance measure using OBDD. The first algorithm is a two-pass traversal algorithm, which computes system reliability in each traversal by setting the reliability of the corresponding component as one in the first travel and zero in the second traversal. The second algorithm is a single-pass traversal algorithm, but has two phases of computation. In the first phase, the probability of path selection from the root node to a specific node $i$ is calculated. In the second phase, the probabilities of its left and right child nodes are calculated. Finally, these probabilities are used to compute the importance measure. Therefore, the algorithms presented in [19] take more computational time. Moreover, this algorithm can only be applied to cases where there is only one path between the root and each node of the OBDD; otherwise, some calculations have to be modified. Later, Ou and Dugan [35] proposed an equation that can produce the importance measure of a component during the reliability evaluation traversal. Therefore, in a single-pass traversal, both the reliability and importance measures of a single component can be evaluated. However, they did not present the implementation of their single-pass traversal. Further, a modular approach to calculate system-level importance measure using module-level importance measure is presented in [35], which can be used to reduce the computational time. However, the approaches presented in both [19] and [35] calculate only one component's importance measure at a time. They need to be run again to obtain the importance measure of another component.

In this section, we propose an algorithm to compute multiple components' importance measures with only a single OBDD traversal. First, we present the importance measures for systems with perfect fault-coverage. Then, we extend our algorithms to evaluate the importance measures of systems with imperfect fault-coverage. Our algorithm can be combined with the modular approach presented in [35] to find the importance of modular fault trees or networks in an efficient way.

### 5.1 Reliability Importance Measures under Perfect Coverage Model

#### 5.1.1 Birnbaum Importance Measure

The Birnbaum importance measure of a component (say component *k*) represents the probability that a system is in a critical state with respect to that component, i.e., the probability that the system is initially in a good state and the failure of component *k* causes the system to fail. It is defined as the partial derivative of system unreliability with respect to the failure probability of component *k*:

$$\begin{aligned} I_k^B(t) &\equiv \frac{\partial F(t)}{\partial F_k(t)} \equiv \frac{\partial R(t)}{\partial R_k(t)} \\ &= \Pr\{g_{y_k=1}\} - \Pr\{g_{y_k=0}\} \\ &= \Pr\{f_{x_k=1}\} - \Pr\{f_{x_k=0}\}, \end{aligned} \tag{26}$$

where $F(t)$ is the system failure probability at time $t$, $F_k(t)$ is the failure probability of component $k$ at time $t$; $F_k(t) = \Pr\{g_{y_k=1}\}$, $g$ is the system structure function, and $f$ is the dual of g, i.e.,

$$\begin{aligned} f(x_1, x_2, \cdots, x_n) &\equiv 1 - g(1 - x_1, 1 - x_2, \cdots, 1 - x_n) \\ &\equiv 1 - g(y_1, y_2, \cdots, y_n), \end{aligned}$$

where $y_i = 1 - x_i$ and $x_i = 1$ (0) means component $i$ is good (faulty). Here, $\Pr\{g_{y_k=1}\}$ is the unreliability of the system given that component $k$ has failed. Similarly, $\Pr\{f_{x_k=1}\}$ is the reliability of the system given that component $k$ is working.

#### 5.1.2 Two-Pass Traversal

This method needs to traverse the OBDD twice to obtain the importance measure of component *k* [19].

- Find $\Pr\{f_{x_k=1}\}$ using OBDD, i.e., find the system reliability by assuming the reliability of component *k* to be 1.
- Find $\Pr\{f_{x_k=0}\}$ using OBDD, i.e., find the system reliability by assuming the reliability of component *k* to be 0.
- $\Pr\{f_{x_k=1}\} - \Pr\{f_{x_k=0}\}$ gives the Birnbaum importance measure of component *k*.

#### 5.1.3 Modified Single-Pass Traversal

This method needs to traverse the OBDD only once to get the importance measure of component *k*. There are two steps. In the first step, from (26), the importance measure depends on the probability of state transition of component *k*. Therefore, a disjoint path, which goes to the terminal one and does not include component *k* in it, will not contribute to the importance measure of component *k*. We should delete this kind of path or let the probabilities of the paths be 0 during traversing the OBDD.

The second step is similar to the procedure in the above section for nodes in finding $\Pr\{f_{x_k=1}\}$ and $\Pr\{f_{x_k=0}\}$ except for the node corresponding to component $k$. At the node corresponding to component $k$, since the reliability of component $k$ in finding $\Pr\{f_{x_k=1}\}$ is 1, the probability of this node is equivalent to the probability of the right subtree. Similarly, since the reliability of component $k$ in finding $\Pr\{f_{x_k=0}\}$ is 0, the probability of this node is equivalent to the probability of the left subtree. Therefore, we combine the two calculations at the node corresponding to component *k* and compute the probability of each node (say *i*) in OBDD using the following rules:

- If node *i* is corresponding to component *k*, then

$$\Pr\{f_k\} = \Pr\{f_{x_k=1}\} - \Pr\{f_{x_k=0}\}.$$

- If node *i* is not corresponding to component *k* and ordering(*i*) > ordering(*k*), then

$$\begin{aligned} \Pr\{f_i\} = \Pr\{x_i\} \cdot \Pr\{f_{x_i=1}\} + (1 - \Pr\{x_i\}) \\ \cdot \Pr\{f_{x_i=0}\}. \end{aligned} \tag{27}$$

- If node *i* is not corresponding to component *k* and ordering(*i*) < ordering(*k*), then also use (27) to calculate $\Pr\{f_i\}$ except that if the right (left) subtree is independent of component *k*, then let $\Pr\{f_{x_i=1}\}$ ($\Pr\{f_{x_i=0}\}$) be 0 in (27). To check if the subtree of node *i* is independent of component *k* is simple. Let node *j* be the subnode of node *i*. If ordering(*j*) > ordering(*k*) then the subtree is independent of component *k*.

Finally, when we have finished traversing the OBDD, we get the probabilityof the root, $\Pr\{f\}$. $\Pr\{f\}$ gives the Birnbaum importance measure of component *k*.

#### 5.1.4 Single-Pass Traversal for Multiple Components

This method needs to traverse the OBDD only once to get the importance measures of multiple components. This method is extended from the method in Section 5.1.3. If $\Omega$ is the set of components whose Birnbaum importance measures are to be calculated, $\Pr\{f(0)\}$ is the system reliability, and $\Pr\{f(k)\}$ is the Birnbaum importance measure of component *k*, then we have the following at each node (say *i*):

- For each node *i*,

$$\begin{aligned} \Pr\{f_i(0)\} = \Pr\{x_i\} \cdot \Pr\{f_{x_i=1}(0)\} + (1 - \Pr\{x_i\}) \\ \cdot \Pr\{f_{x_i=0}(0)\}. \end{aligned}$$

- For each $k \in \Omega$,
  - If node *i* is corresponding to component *k*, then

$$\Pr\{f_i(k)\} = \Pr\{f_{x_i=1}(0)\} - \Pr\{f_{x_i=0}(0)\}$$

  - If node *i* is not corresponding to component *k* and ordering(*k*) > ordering(*i*), (i.e. $\Pr\{f_{x_k=1}(k)\}$ and $\Pr\{f_{x_k=0}(k)\}$ have been calculated), then let $\Pr\{f_{x_k=1}(k)\}$ ($\Pr\{f_{x_k=1}(k)\}$) be 0 in (28) if the right (left) subtree is independent of component $k$. Then, calculate $\Pr\{f_i(k)\}$ using (28).

$$\begin{aligned} \Pr\{f_i(k)\} = \Pr\{x_i\} \cdot \Pr\{f_{x_i=1}(k)\} + (1 - \Pr\{x_i\}) \\ \cdot \Pr\{f_{x_i=0}(k)\}. \end{aligned} \tag{28}$$

```
struct imp {          // Importance Measure
    double f[m];   // the set of components whose Birnbaum's importance
}                     // measures is to be evaluated
main() {
    static imp BI, OneArray, ZeroArray;
    for k = 0 to m
        OneArray.f[k] = 1;
        ZeroArray.f[k] = 0;
    next
    BI = measure(BDDnode root);
    //   BI.f[0] is the system reliability;
    //   BI.f[k] is the Birnbaum's importance measure of component k;
}
Procedure imp measure(BDDnode x_i) {        // Birnbaum's Importance Measure
    imp result, n_true, n_false;
    if (x_i = BDD_one ) then return OneArray;
    if (x_i = BDD_zero ) then return ZeroArray;
    if ( result = get_computed_table(x_i) is a hit ) then return (result);
    n_true = measure(sub_node_true(x_i));
    n_false = measure(sub_node_false(x_i));
    result.f[0] = p * n_true.f[0] + q * n_false.f[0];         // q = 1 – p
    for k = 1 to m                                              // the set of components
        if (x_i is corresponding to component k ) then         // whose  Birnbaum's importance
            result.f[k] = n_true.f[0] – n_false.f[0];           // measure is to be evaluated
        elseif ( ordering(x_i) < ordering(k) ) then
            prob_sub_true = n_true.f[k];
            prob_sub_false = n_false.f[k];
            if n_true is independent of component k then prob_sub_true = 0;
            if n_false is independent of component k then prob_sub_false = 0;
            result.f[k] = p * prob_sub_true + q * prob_sub_false;
        end if
    next
    insert_computed_table(x_i, result);
    return (result);
}
```

Fig. 14. The OBDD-based algorithm for the calculation of the Birnbaum importance measure.

- Otherwise, do nothing since $\Pr\{f_i(k)\}$ is equivalent to $\Pr\{f_i(0)\}$.

Finally, the probabilities, $\Pr\{f(k)\}$ for all $k \in \Omega$, at the root in the OBDD gives the Birnbaum importance measure of component *k*. Fig. 14 illustrates the OBDD-based algorithm for the calculation of Birnbaum importance measures of multiple components by traversing the OBDD only once.

#### 5.1.5 Criticality Importance Measure

The criticality importance measure of component *k* of a system is defined as the probability that component *k* becomes faulty at time *t* and, at the same time, is critical to the occurrence of system failure, given that the system has failed.

$$I_k^{Cr}(t) \equiv \frac{\partial F(t)}{\partial F_k(t)} \times \frac{F_k(t)}{F(t)} \equiv I_k^B(t) \times \frac{F_k(t)}{F(t)}. \tag{29}$$

It should be noted that the criticality importance measure of component *k* could be calculated directly from (29) given that we have the Birnbaum importance measure of component *k*. The Birnbaum importance measure can be calculated using the algorithm presented in Section 5.1.3. If we want to evaluate this measure for all components, then a similar algorithm that is used for the Birnbaum importance measure in Section 5.1.4 can be used for this case.

#### 5.1.6 Risk Reduction Ratio

This ratio indicates decrease in system unreliability when the unreliability of a given component, say component *k*, is zero (i.e., component *k* never fails). This ratio is defined as

$$RRR_k(t) \equiv \frac{F(t)}{F_{x_k=1}(t)} \equiv \frac{1-R(t)}{1-R_{x_k=1}(t)} = \frac{1-\Pr\{f\}}{1-\Pr\{f_{x_k=1}\}}.$$

The evaluation of this measure is similar to the evaluation of Birnbaum importance measure. We first find $\Pr\{f\}$ and then find $\Pr\{f_{x_k=1}\}$. In fact, using the concept of Section 5.1.4, we can get the result by traversing the OBDD only once.

#### 5.1.7 Risk Reduction Interval

This index is similar to the Risk Reduction Ratio except that it uses the actual unreliability difference instead of the unreliability ratio. It is defined as

$$\begin{aligned} RRI_k(t) &\equiv F(t) - F_{x_k=1}(t) \equiv R_{x_k=1}(t) - R(t) \\ &= \Pr\{f_{x_k=1}\} - \Pr\{f\}. \end{aligned}$$

However, we make some modifications and get

$$\begin{aligned} RRI_k(t) &\equiv \Pr\{f_{x_k=1}\} - (\Pr\{x_k\}\cdot\Pr\{f_{x_k=1}\} + (1-\Pr\{x_k\}) \\ &\quad \cdot \Pr\{f_{x_k=0}\}) \\ &= (1-\Pr\{x_k\})\cdot\Pr\{f_{x_k=1}\} - (1-\Pr\{x_k\}) \\ &\quad \cdot \Pr\{f_{x_k=0}\} \\ &= (1-\Pr\{x_k\})\cdot(\Pr\{f_{x_k=1}\} - \Pr\{f_{x_k=0}\}). \end{aligned}$$

Therefore, the evaluation of this measure is similar to the evaluation of the Birnbaum importance measure by traversing the OBDD once, as discussed in Section 5.1.4.

#### 5.1.8 Risk Increase Ratio

This ratio indicates the increase in the system unreliability when the unreliability of a given component, say component *k*, is one (i.e.. component *k* always fails). This ratio is defined as

$$RIR_k(t) \equiv \frac{F_{x_k=0}(t)}{F(t)} \equiv \frac{1-R_{x_k=0}(t)}{1-R(t)} = \frac{1-\Pr\{f_{x_k=0}\}}{1-\Pr\{f\}}.$$

Evaluation of this measure is similar to the evaluation of the Risk Reduction Ratio. Using the concept of Section 5.1.4, we can get the result by traversing the BDD only once.

#### 5.1.9 Risk Increase Interval

This index is similar to the Risk Increase Ratio; however, it uses the actual unreliability difference instead of the unreliability ratio. It is defined as

$$\begin{aligned} RII_k(t) &\equiv F_{x_k=0}(t) - F(t) \equiv R(t) - R_{x_k=0}(t) \\ &= \Pr\{f\} - \Pr\{f_{x_k=0}\} \\ &= \Pr\{x_k\}\cdot(\Pr\{f_{x_k=1}\} - \Pr\{f_{x_k=0}\}). \end{aligned}$$

The evaluation of this measure is similar to the evaluation of the Risk Reduction Interval.

### 5.2 Reliability Importance Measures under Imperfect Coverage Model

This section presents several algorithms to compute the importance measures for a system with imperfect fault-coverage. For simplification, we consider the nonrepairable system and the importance measures are calculated assuming no change in the coverage factor with respect to the change in the component failure probability (i.e.. for fixed coverage factors).

#### 5.2.1 Birnbaum Importance Measure

From (25) as well as [21], the reliability of a system with imperfect coverage is as follows:

$$R = P_u \cdot R_c, \tag{30}$$

where $P_u$ is the probability of no uncovered failure in the system and $R_c$ is the conditional reliability of the system given that no uncovered failure has occurred in this system.

$$P_u = \prod_{i=1}^{n} [r_i + (1 - r_i)c_i],$$

where $r_i$ is the reliability of component $i$ and $c_i$ is the coverage factor of component $i$.

The conditional reliability of component $i$, $R_i$, given that no uncovered failure has occurred, is

$$R_i = \frac{r_i}{r_i + (1 - r_i)c_i}.$$

By (30), we have

$$\frac{\partial R}{\partial r_k} = R_c \frac{\partial P_u}{\partial r_k} + P_u \frac{\partial R_c}{\partial r_k},$$

where

$$\begin{aligned}\frac{\partial P_u}{\partial r_k} &= \prod_{i=1, i \neq k}^{n} [r_i + (1 - r_i)c_i] \cdot (1 - c_j) = P_u \frac{1 - c_k}{r_k + (1 - r_k)c_k} \\ &= P_u(1 - c_k)\frac{R_k}{r_k},\end{aligned}$$

and, by the chain rule of differentiation,

$$\begin{aligned}\frac{\partial R_c}{\partial r_k} &= \frac{\partial R_c}{\partial R_k} \cdot \frac{\partial R_k}{\partial r_k} \\ \frac{\partial R_k}{\partial r_k} &= \frac{r_k + (1 - r_k)c_k - r_k(1 - c_k)}{(r_k + (1 - r_k)c_k)^2} = c_k \left(\frac{R_k}{r_k}\right)^2.\end{aligned}$$

Therefore,

$$\begin{aligned}\frac{\partial R}{\partial r_k} &= R_c P_u (1 - c_k)\frac{R_k}{r_k} + P_u \frac{\partial R_c}{\partial R_k} c_k (\frac{R_k}{r_k})^2 \\ &= P_u \frac{R_k}{r_k}\left[(1 - c_k)R_c + c_k \frac{\partial R_c}{\partial R_k}\frac{R_k}{r_k}\right].\end{aligned} \tag{31}$$

In order to derive $\partial R / \partial r_i$, we need to find only $\partial R_c / \partial R_k$ because all other parameters are known already. Finding $\partial R_c / \partial R_k$ is equivalent to finding the Birnbaum importance measure of the system with perfect coverage, but with the modified values of the component reliabilities, that is, the conditional reliability of component $i$, $R_i$. The Birnbaum importance measure can be evaluated using the method in Section 5.1.3.

Therefore, from (26), the Birnbaum importance measure under the imperfect coverage model is

$$I^B_{k\_ICPM}(t) \equiv \frac{\partial R(t)}{\partial r_k(t)}.$$

The calculation of this measure is straightforward from (31).

#### 5.2.2 Criticality Importance Measure

From the definition and (29), the criticality importance measure under the imperfect coverage model is

$$I^{Cr}_{k\_IPCM}(t) \equiv I^B_{k\_IPCM}(t) \times \frac{F_k(t)}{F(t)}.$$

Therefore, the calculation of this measure is simple given that we have the Birnbaum importance measure of component *k*.

#### 5.2.3 Risk-Based Measures

The evaluation of the risk-based measures for the imperfect coverage case is similar to the case of perfect coverage except that the effect of imperfect coverage should be taken into account in evaluating the system unreliability.

## 6 Conclusions

We have presented a new efficient algorithm based on OBDD for the calculation of the time-specific as well as the steady-state failure frequency of a system. This algorithm can also be applied to the case of global time-dependent failure and repair rates if the individual component availability is calculated using appropriate techniques. In addition, we have proposed an OBDD-based algorithm for the evaluation of importance measures, including the Birnbaum importance, the Criticality importance, and other indices for risk evaluation of systems. Moreover, all of the proposed algorithms based on OBDD in this paper could also be extended to analyze a system with imperfect fault-coverage.

The imperfect coverage model (IPCM) is important to accurate reliability assessment of a fault-tolerant computer system. We have also proposed an approach for incorporating the IPCM of a repairable system into a combinatorial model. The model using Markov chains for repairable systems with imperfect fault-coverage is quite useful. Due to the use of conditional probabilities and Markov chains, our OBDD-based algorithm is very efficient for the reliability evaluation of a nonrepairable system and the availability evaluation of a repairable system with imperfect fault-coverage.

In this paper, the powerful capability of OBDD for reliability evaluation has been fully exploited with the proposed algorithms. These OBDD-based algorithms are very efficient in both computational time and storage demand for reliability analysis and also make it possible for us to study practical and large distributed systems. Based on the approaches, researches on sensitivity analysis, importance measures, failure frequency analysis, or optimal design issues of distributed systems and multistate systems will be the focus of our future works.

## Acknowledgments

This research was supported by the National Science Council, Taiwan, Republic of China, under Grant NSC 92-2623-7-002-004-NU.

## References

[1] A. Rauzy, "A New Methodology to Handle Boolean Models with Loops," *IEEE Trans. Reliability*, vol. 52, no. 1, pp. 96-105, Mar. 2003.

[2] J.D. Andrews and S.J. Dunnett, "Event-Tree Analysis Using Binary Decision Diagrams," *IEEE Trans. Reliability,* vol. 49, pp. 230-338, June 2000.

[3] 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.

[4] W.G. Schneeweiss, *The Fault Tree Method.* Hagen: LiLoLe-Verlag, 1999.

[5] L. Xing and J.B. Dugan, "Reliability Analysis of Phased-Mission Systems with Combinatorial Phase Requirements," *Proc. Ann. Reliability and Maintainability Symp. (RAMS '01),* pp. 344-351, 2001.

[6] L. Xing and J.B. Dugan, "Analysis of Generalized Phased-Mission Systems Reliability, Performance, and Sensitivity," *IEEE Trans. Reliability,* pp. 199-211, June 2002.

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

[8] H. Kumamoto and E.J. Henley, *Probabilistic Risk Assessment and Management for Engineers and Scientists,* second ed. IEEE Press, 1996.

[9] K.B. Misra, *Reliability Analysis and Prediction: A Methodology Oriented Treatment.* Elsevier, 1992.

[10] M. Veeraraghavan and K.S. Trivedi, "Multiple Variable Inversion Techniques," *New Trends in System Reliability Evaluation,* K.B. Misra, ed., pp. 39-74, Elsevier, 1993.

[11] S.V. Amari, "Generic Rules to Evaluate System-Failure Frequency," *IEEE Trans. Reliability,* vol. 49, pp. 85-87, Mar. 2000.

[12] R. Billinton and S. Jonnavithula, "Calculation of Frequency, Duration, and Availability Indexes in Complex Network," *IEEE Trans. Reliability,* vol. 48, pp. 25-30, Mar. 1999.

[13] C. Singh, "Calculating the Time-Specific Frequency of System Failure," *IEEE Trans. Reliability,* vol. 28, pp. 124-126, June 1979.

[14] C. Singh, "Rules for Calculating the Time-Specific Frequency of System Failure," *IEEE Trans. Reliability,* vol. 30, pp. 364-366, Oct. 1981.

[15] C. Singh and R. Billinton, *System Reliability Modelling and Evaluation.* Hutchinson & Co., 1977.

[16] W.G. Schneeweiss, "Limited Usefulness of BDDs for Mean Failure Frequency Calculation," *J. Automatic Control Production Systems,* pp. 1131-1136, 1996.

[17] W.G. Schneeweiss, "Advanced Fault Tree Modeling," *J. Universal Computer Science,* vol. 5, pp. 633-643, 1999.

[18] W.G. Schneeweiss, "Fast Fault-Tree Evaluation for Many Sets of Input Data," *IEEE Trans. Reliability,* vol. 39, pp. 296-300, Aug. 1990.

[19] R. Sinnamon and J.D. Andrews, "Fault Tree Analysis and Binary Decision Diagrams," *Proc. Ann. Reliability and Maintainability Symp. (RAMS '96),* pp. 215-222, Jan. 1996.

[20] W.G. Schneeweiss, "The Failure Frequency of Systems with Dependent Components," *IEEE Trans. Reliability,* vol. 35, pp. 512-517, Dec. 1986.

[21] 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.

[22] 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.

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

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

[25] 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.

[26] J.M. Nahman, "Failure-Frequency Evaluation of Complex Systems Using Cut-Set Approach," *IEEE Trans. Reliability,* vol. 30, pp. 353-355, Oct. 1981.

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

[28] H. Pham, "Optimal Cost-Effective Design of Triple-Modular-Redundancy-with-Spares Systems," *IEEE Trans. Reliability,* vol. 42, pp. 369-374, Sept. 1993.

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

[30] J.B. Dugan and K.S. Trivedi, "Coverage Modeling for Dependability Analysis of Fault-Tolerant Systems," *IEEE Trans. Computers,* vol. 38, no. 6, pp. 755-787, June 1989.

[31] S. Akhtar, "Reliability of *k*-out-of-*n*:G Systems with Imperfect Fault-Coverage," *IEEE Trans. Reliability,* vol. 43, pp. 101-106, Mar. 1994.

[32] L. Yin, M.A.J. Smith, and K.S. Trivedi, "Uncertainty Analysis in Reliability Modeling," *Proc. Ann. Reliability and Maintainability Symp. (RAMS '01),* pp. 229-234, 2001.

[33] H. Ayhan, J. Limon-Robles, and M. Wortman, "An Approach for Computing Tight Numerical Bounds on Renewal Functions," *IEEE Trans. Reliability,* vol. 39, pp. 182-188, June 1999.

[34] E. Smeitink and R. Dekker, "A Simple Approximation to Renewal Function," *IEEE Trans. Reliability,* vol. 39, pp. 71-75, Apr. 1990.

[35] Y. Ou and J.B. Dugan, "Sensitivity Analysis of Modular Dynamic Fault Trees," *Proc. Computer Performance and Dependability Symp. (IPDS '00),* pp. 35-43, 2000.


**Yung-Ruei Chang** received the MS degree (1995) in electrical engineering from National Taiwan University. Since 1996, he has been working as an assistant researcher at the Institute of Nuclear Energy Research, Atomic Energy Council, Taiwan. He has been pursuing the PhD degree in electrical engineering from National Taiwan University since 2000. His research interests include fault-tolerant system reliability, distributed dependable system reliability, network reliability, and dependable computing. He is a student member of the IEEE.

**Suprasad V. Amari** received the BS dgree (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 reliability software design rngineer at Relex Software Corporation. In his work with Relex Software Corporation, he is involved in designing the Relex Markov module, the Relex System Optimization and Simulation (OpSim) module, Optimal Scheduling and Replacement Policies, Test Planning, Reliability Growth Models, Dynamic Fault Trees, and algorithms to compute the reliability of systems with repairable components. 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 is a member of the IEEE, ACM, and ASQ.

**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. Since 1991, he has been with National Taiwan University, where he is currently a professor and the chairman of Department of Electrical Engineering. He spent his sabbatical year as a visiting researcher at AT&T Labs-Research, New Jersey from 1999 to 2000. He was the chairman of the Department of Computer Science and Information Engineering, 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 the California Institute of Technology. His current research interests include software reliability engineering, mobile computing, dependable systems and networks, and optical WDM networks. He is an IEEE fellow and a member of the IEEE Computer Society. He has published more than 180 papers in journals and conferences.