Reliability of demand-based phased-mission systems subject to fault level coverage

Rui Peng a, Qingqing Zhai b, Liudong Xing a,*, Jun Yang b

a Dongling School of Economics and Management, University of Science and Technology Beijing, Beijing, China
b School of Reliability and Systems Engineering, Beihang University, Beijing, China

ARTICLE INFO

Article history:
Received 19 March 2013
Received in revised form 21 July 2013
Accepted 27 July 2013
Available online 2 August 2013

Keywords:
Phased-mission system
Reliability evaluation
Multi-valued decision diagram
Mission demand
Fault level coverage

ABSTRACT

In many real-world applications, a mission may consist of several different tasks or phases that have to be accomplished in sequence. Such systems are referred to as phased-mission systems (PMS). In this paper we consider the demand-based PMS with parallel structure, where the system components function in parallel with different capacities in each phase of the mission and the mission is successful if and only if the total system capacity meets the predetermined mission demand in each phase. The reliability of the demand-based PMS (DB-PMS) with parallel structure subject to fault-level coverage (FLC) is first studied using a multi-valued decision diagram (MDD) based technique. The traditional MDD is modified to accommodate the FLC mechanism and new MDD construction and evaluation procedures are proposed for DB-PMS. To reduce the size of the MDD, an alternative construction procedure applying the branching truncation method and new reduction rules are further proposed. An upwards algorithm is put forward to evaluate the reliability of DB-PMS subject to FLC. The proposed approaches are illustrated through examples.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

In many real-world applications, a mission may consist of several different tasks or phases which have to be accomplished in sequence [1]. For instance, an aircraft flight involves taxi, take-off, ascent, level-flight, descent and landing phases [2]. Such systems are referred to as phased-mission systems (PMS). During each phase, the system has to accomplish a specified task and may suffer different environment or stress conditions. Thus, the system configuration, success criteria, reliability requirements, and component behavior may vary from phase to phase [3–5]. For example, for a twin-engine aircraft, a single engine can ensure the function of the taxi phase whereas both engines are required during the take-off phase. Moreover, the engines usually suffer higher stress and are more likely to fail during the take-off phase than other phases.

Compared with single-phase systems, reliability analysis of PMS is more complex due to the above mentioned system dynamics as well as dependence across the phases (the state of a component at the beginning of a new phase should be identical to its state at the end of the previous phase) [6–8]. Generally, approaches to analyzing PMS can be classified into two classes: analytical methods and simulations [9]. The simulation methods are flexible and can be easily modified to adapt to different situations. However, they are computationally inefficient and require considerable numbers of runs to evaluate the system reliability. The analytical methods can explicitly reflect the system structure and the results might be used in further applications, e.g. optimization of the system design. The analytical methods can be further classified into state-space oriented models [10–12], combinatorial methods [13,14] and a phase modular solution [15] that combines the former two methods. The state-space oriented models (in particular Markov or Petri nets based methods) could handle both static and dynamic PMS. However they are difficult to be applied to large-scale systems due to the well-known state-space explosion problem. The combinatorial methods exploit the Boolean algebra and decision diagrams to reduce the computational complexity, which makes them applicable to handle larger-scale systems.

This paper focuses on the decision diagram based combinatorial methods. A general “demand-based” phased-mission system (DB-PMS) with parallel structure is considered. Similar to studies on capacitated networks [16,17] and weighted k-out-of-n systems [18–20], it is assumed that the system capacity is the sum of the capacities of the working components. In order to adapt to a wider range of applications, the component capacity is allowed to change from phase to phase. For example, the memory of chips can change when the heat or current changes; the capacity of solar panels can

© 2013 Elsevier Ltd. All rights reserved.
also change according to the environment. The required demand for each phase of the mission is predetermined; the mission succeeds if and only if the system capacity exceeds the predetermined demand in all phases. For example, in a power system, each power plant has a nominal capacity and the total capacity of the working power plants determines the available electricity. The demand for the electricity varies in different periods of a day, and the power system is considered available if the demand is met during the whole day. To ensure the success and high reliability of the mission, redundancy has to be consumed especially for safety-critical or life-critical PMS; the redundancy is consumed with the failure of the component and the entire mission fails if the total capacity of the remaining components cannot meet the system demand in any phase.

For systems designed with redundancy, it may happen that a single uncovered component fault propagates through the system and leads to the overall system failure even when the remaining redundancy is still adequate [21,22]. Consider a scientific computation system where several processors are used simultaneously. If one processor fails without being detected or isolated, it may produce incorrect result which can lead to the corruption of the whole computation. This behavior is known as imperfect fault coverage (IFC) [23,24]. The system reliability may not increase unlimithedly with the increase of the system redundancy when IFC is considered [25–27]. According to the fault covering mechanism, two types of IFC models have been studied in the literature: element level coverage (ELC) and fault level coverage (FLC) [28]. In the ELC model, each component is supposed to fail with a specified coverage factor, that is, the failure of the component is covered with a probability $c$ and not covered with probability $1-c$. In the FLC model, the first component failure has a probability $c_1$ of being covered, the second one has a probability $c_2$ of being covered, and so on. While most of the existing works on PMS analysis did not consider the IFC effect at all [3,5,8,12,14], some researchers studied the reliability of PMS subject to ELC [1,2,4,6]. FLC has been studied for single-phase systems [28,29], however, to the best of our knowledge, no work has been performed to consider the effect of FLC in the reliability analysis of PMS. In this paper, the DB-PMS subject to FLC is first analyzed and multi-valued decision diagram (MDD) is adapted to evaluate the system reliability.

The rest of the paper is organized as follows. Section 2 gives an overview of the problem to be solved including the system description and model assumptions. Section 3 presents the MDD based approach for the evaluation of the parallel DB-PMS reliability. Section 4 focuses on the complexity issues, including an alternative way to construct the MDD, the MDD reduction procedure, the upwards algorithm, and the computational complexity analysis. Examples are given in Section 5 to illustrate the proposed method. Conclusions are given in the end.

2. Problem statement

The DB-PMS with parallel structure subject to FLC is studied in this paper. The description and assumptions of the system are:

(1) The system mission consists of $M$ consecutive non-overlapping phases. The duration of each phase is deterministic.

(2) The system consists of $n$ components which are statistically independent but not necessarily identical. Each component $i$ in phase $j$ ($1 \leq j \leq M$) has a nominal capacity $w_{ij}(1 \leq i \leq n, 1 \leq j \leq M)$ and the system capacity in phase $j$ is the summation of the capacities of the working components in phase $j$.

(3) The system demand may vary from phase to phase. To accomplish the mission successfully, the system has to meet the predetermined demand $d_j(1 \leq j \leq M)$ in each phase $j$.

(4) For each component, it may suffer different stresses or environment conditions in different phases, which lead to different failure behaviors in different phases. In this paper, the accelerated life time model [30,31] is utilized to model the components’ failure behavior under different stresses.

3. MDD-based approach

3.1. Traditional MDD

The MDD method provides an efficient and exact way to analyze static multi-state fault trees [32,33]. The traditional MDD...
has two sink nodes, labeled ‘0’ and ‘1’, representing a binary-state system being in the operational and failed states respectively. For a system with multiple system states, these two sink nodes represent the system being not in or in a particular system state, respectively. A non-sink node in MDD has multiple outgoing edges each corresponding to a state of the multi-state component represented by the node. In particular, a non-sink node representing a multi-state component \( A \) with \( k \) states is associated with an \( k \)-value state variable \( x_k \) and has \( k \) outgoing edges \( \{1,2,...,k\} \). The logic expression for \( A \) can be represented in MDD using the case format as:

\[
F = A_1 \times F_{x_1} + A_2 \times F_{x_2} + \cdots + A_k \times F_{x_k} = \text{case}(A, F_1, F_2, ..., F_k)
\]  

(1.1)

All the sub-expressions \( A_l \times F_{x_l} \) in (1) are disjoint. The MDD can be generated recursively by applying manipulation rules of (2) [32]:

\[
G \circ H = \text{case}(x, G_1, ..., G_n) \circ \text{case}(y, H_1, ..., H_k)
\]

\[
= \begin{cases} 
\text{case}(x, G_1 \circ H_1, ..., G_n \circ H_k) \text{ if } \text{index}(x) = \text{index}(y) \\
\text{case}(x, G_1 \circ H_1, ..., G_n \circ H_k) \text{ if } \text{index}(x) < \text{index}(y) \\
\text{case}(x, G_1 \circ H_1, ..., G_n \circ H_k) \text{ if } \text{index}(x) > \text{index}(y)
\end{cases}
\]  

(2)

where \( G \) and \( H \) represent two multi-valued expressions corresponding to the traversed sub-fault trees. The logical operation (AND, OR) is represented by \( \circ \).

The failure probability of the system (or the probability of the system being in a particular state) can be calculated as the sum of probabilities of all the disjoint paths from the root node to the sink node ‘1’. These paths represent all possible combinations of component states that lead to the entire system failure (or the system being in a particular state).

### 3.2. MDD-based method for DB-PMS

Before the construction of the system level MDD, the MDD representation of a single component has to be built first. Since we are considering the DB-PMS, the system capacity during each phase should be represented by the MDD; moreover, for considering FLC, the number of failed components should also be incorporated into the MDD construction. In the light of these considerations, each component can be represented with a MDD incorporating the components, the duration of each phase and the corresponding failure probabilities of all the disjoint paths from the root node to the sink node ‘1’. These paths represent all possible combinations of component states that lead to the entire system failure (or the system being in a particular state).

<table>
<thead>
<tr>
<th>Terminal value</th>
<th>MDD representation of component</th>
<th>MDD representation of system</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( A_1 )</td>
<td>( A_1 \circ A_2 )</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>( A_1 \circ A_2 )</td>
</tr>
</tbody>
</table>

Consider a two-phase system of two components \( A_1 \) and \( A_2 \) with capacities \( w_{11} = 3 \) and \( w_{21} = 2 \) respectively. The MDD representations of \( A_1 \) and \( A_2 \) are given as Fig. 2(a) and (b) respectively. The system level MDD can be constructed according to the proposed algorithm. First, the construction starts with the MDD representation of component \( A_1 \). Then the MDD of \( A_2 \) is added to each terminal of \( A_1 \) and the new terminal value equal to the sum of the terminal value of \( A_1 \) and the terminal value of \( A_2 \). Fig. 2(c) gives the MDD for this system.

After the construction of the system MDD, the system reliability can be obtained straightforward. Given the lifetime distributions of the components, the duration of each phase and the corresponding acceleration factor in each phase, the failure probability of component \( A_i \) (1 ≤ \( i \) ≤ \( n \)) during phase \( 1 \leq j \leq M \) can be obtained as:

\[
q_{ij} = F_i(\sum_{k=1}^{j-1} \alpha_k T_k) - F_i(\sum_{k=1}^{j-1} \alpha_k T_k)
\]

(3)

The probability that the component does not fail in any phase is \( p_i = 1 - q_i \) for each \( 1 \leq i \leq n \) and \( 1 \leq j \leq M \).
and \( p_i \) of each component on the path. The occurrence probability of the path considering FLC is

\[
R(\text{path}) = Pr(\text{path}) \times c_1 \times \cdots \times c_{N_{\text{path}}},
\]

(3)

Finally the system reliability can be obtained as the summation of the occurrence probabilities of all the paths whose terminal value \( C_{\text{path}} \) is not smaller than \( D \), i.e. the demands are satisfied in all \( M \) phases. In the preceding example, suppose the demand of the mission is \((4,2)\), then the system reliability based on the MDD of Fig. 2(c) is

\[
R_1 = Pr(\text{path } 1) + c_1(Pr(\text{path } 3) + Pr(\text{path } 7)) = p_1p_2 + c_1(p_1q_22 + q_12p_2)
\]

(4)

4. Complexity issues

With the construction algorithm in Section 3.2, the system MDD can be constructed and then the system reliability can be obtained subsequently. However, the system MDD is a complete tree with many unnecessary nodes and branches, which results in a waste of computer memory or construction effort. Hence, we consider the complexity reduction of the algorithm in this section. First, an alternative construction procedure with a branching truncation mechanism is proposed to simplify the construction of the MDD. Second, a reduction procedure is given to further reduce the MDD. Third, an upwards algorithm to evaluate the system reliability is proposed. At last, the computational complexity is analyzed.

4.1. The branching truncation method to construct MDD

The system capacity \( W_s = (\sum_{i=1}^{n} W_{i1}, \ldots, \sum_{i=1}^{n} W_{iM}) \) with all components working in all \( M \) phases is computed first. An auxiliary node with one branch and the terminal value \((0,W_s)\) is constructed to represent the perfect state of the system: the state with no failed components, as shown in Fig. 3(a). Then, the MDD representation of \( A_i (1 \leq i \leq n) \) are modified as shown in Fig. 3(b) for instance, the terminal value of the path “failed in the \( M \)th phase” is \( 1_{(0,0,\ldots,w_s)} \), which means that compared with the perfect system state, there is one more failure and a loss of capacity \( w_s \) in the \( M \)th phase due to the failure of component \( A \).

The system MDD can be constructed according to the following iterative algorithm:

1) Start the MDD construction with Fig. 3(a).
2) Add the MDD representation of \( A_i (1 \leq i \leq n) \) to every terminal where the current terminal value \( C_{\text{path}} \) is not smaller than the mission demand \( D \) (since the new added MDD representation will not increase the system capacity, there is no need to develop this path any further if the current terminal value \( C_{\text{path}} \) is smaller than the mission demand \( D \)). The new terminal value \( N_{\text{path}} \) equals to the original terminal value plus the terminal value of the MDD representation of \( A_i \) i.e. \( I_i \) whereas the new terminal value \( C_{\text{path}} \) equals to the original value minus the terminal value of the MDD representation of \( A_i \) i.e. \( W_i \). That is,

\[
N_{\text{path}} = N_{\text{path}} + I_i; C_{\text{path}} = C_{\text{path}} - W_i.
\]

Applying this alternative construction algorithm to the example in Section 3.2, the auxiliary node, the corresponding MDD representations for \( A_1, A_2 \) and the system MDD are given in Fig. 4.

Comparing Fig. 4(d) and Fig. 2(c), it is obvious that the paths that correspond to the system success are identical (except for the auxiliary node) but the size of the system MDD is reduced.

4.2. Further reduction of the MDD

In general, since the capacities of the components may be different, the capacity vectors \( C_{\text{path}} \) in the sink nodes are different with each other. Therefore, the MDD cannot be reduced using the traditional reduction rule of merging isomorphic sub-trees. However, as we are only concerned with the mission being successful or not; the actual system capacity in each phase does not matter, so the constructed MDD can be transformed and simplified in the following way.

1) If \( C_{\text{path}} \) exceeds the demand \( D \), use \( N_{\text{path}} \) to substitute the bivariate \( (N_{\text{path}}, C_{\text{path}}) \) in all the sink nodes; else use “-1” to represent the failure of the system.
2) Merge isomorphic sub-trees and delete the useless nodes where all the outgoing edges leading to a unique child node. Then the new reduced MDD can be obtained.

Considering the illustrative two-phase system, the MDD in Fig. 4(d) after reduction is given in Fig. 5.
3.3. Upwards MDD evaluation algorithm

After the reduction, the system reliability can be calculated with an upwards algorithm. First, we denote the overall coverage factor for a certain path that contributes to the system reliability as

\[ c_{\text{path}} = \prod_{k=1}^{N_{\text{non}}-1} c_k. \]

Then the occurrence probability of a path can be expressed as \( c_{\text{path}} \Pr(\text{path}) \). Before applying the upwards algorithm, we first assign a weight “0” to the “−1” node, a weight “1” to the “0” node, and \( c_{\text{path}} \) to other sink nodes.

From the sink nodes bottom-up with the “breadth-first” traversal, for each node \( A_i \), we calculate the conditional system reliability given that the path from \( A_1 \) to the \( A_i \) node occurs as

\[
\Pr(A_i) = p_i \Pr(\text{the first child node of } A_i) \\
+ \sum_{j=1}^{M} q_{ij} \Pr(\text{the } (j+1)\text{th child node of } A_i)
\]

where \( \Pr(\text{the } j\text{th child node of } A_i) \) is the conditional system reliability given that the path from \( A_1 \) to the \( j\text{th child node of } A_i \) occurs. In particular, in case where the \( j\text{th child node of } A_i \) is a sink node, then \( \Pr(\text{the } j\text{th child node of } A_i) \) is equal to the corresponding weight of the sink node. After traversing from the sink nodes to the top node, the system reliability can be obtained as \( \Pr(A_1) \).

For illustration, we apply the preceding procedure to the reduced MDD in Fig. 5 and then the system reliability is given as \( \Pr(A_1) = p_1(p_2 + c_1q_{12}) + c_1q_{12}p_2 \) (Fig. 6), which is identical to the system reliability expression given in Eq. (4).

4.4. Computational complexity

It is essential to present the computational complexity when an algorithm is proposed [24]. For the MDD constructed with the algorithm in Section 3.2, there are \((M+1)^n\) sink nodes (identical to the number of paths) and \(\sum_{k=0}^{n}(M+1)^k = (M+1)^{n+1}-1)/M \) non-sink nodes, whereas there are at most \(n+1\) sink nodes after reduction and the reduced MDD usually has much fewer non-sink nodes than the complete tree (which becomes clear when referring to Section 5.2). The computational complexity of the upwards algorithm is almost linear with the number of the non-sink nodes (NONSN) in the reduced MDD. So we can expect much computational benefits with the proposed method. It has been proved empirically that the MDD method are more efficient than the exhaustive search method [32,34,35], similar to the binary decision diagram [36,37]. For example, in the illustrative two-phase system, three rounds of calculation (identical to NONSN) can give the system reliability while 9 combinations have to be considered with the exhaustive method.

5. Case studies

5.1. Case 1: DB-PMS with 3 phases and 4 components

To illustrate the proposed methods, we consider a DB-PMS involving 4 components as well as 3 non-overlapping and consecutive phases. The detailed information of the system is given in Table 1.

To illustrate the application of the proposed method on large-scale systems, a DB-PMS with 4 phases and 20 components is studied. The detailed information of the system is given in Table 3.

The probability \( q_{ij} \) that component \( A_i (1 \leq i \leq 4) \) fails in phase \( j (1 \leq j \leq 3) \) and the probability \( p_i \) that component \( A_i \) survives all the 3 phases are obtained and listed in Table 2. The system reliability can be obtained as the summation of the occurrence probabilities of all the paths whose terminal value is not “−1”. Here the system reliability is:

\[
R_s = R_1 + c_1(R_2 + R_3 + R_4 + R_5 + R_6 + R_{12}) \\
+ c_1c_2(R_6 + R_7 + R_8 + R_{10} + R_{11})
\]

which is identical with the result calculated by the upwards method.

5.2. Case 2: DB-PMS with 4 phases and 20 components

To illustrate the application of the proposed method on large-scale systems, a DB-PMS with 4 phases and 20 components is studied. The detailed information of the system is given in Table 3.

The probability \( q_{ij} \) that component \( A_i (1 \leq i \leq 20) \) fails in phase \( j (1 \leq j \leq 4) \) and the probability \( p_i \) that component \( A_i \) survives all the 4 phases are obtained and listed in Table 4. The MDD is constructed and reduced according to Section 4. The system reliability is calculated with the upwards algorithm. Table 5 lists the number of the non-sink nodes (NONSN) in the MDD and the system reliability for the system subject to different mission demands. To show the efficiency of the proposed method, Table 5 also reports the corresponding CPU times on a Core 2 Duo 2.93 GHz for the MDD construction, the MDD reduction and the system reliability calculation with the upwards algorithm. Compared with the complete tree, in which NONSN is \((5^{20} - 1)/4 \approx 2.38 \times 10^{12}\), NONSN in the MDD constructed by the branching truncation algorithm is much smaller. Moreover, NONSN in the MDD after reduction is much smaller than the original one. Hence, the proposed method could significantly simplify the reliability computation.

Fig. 5. The reduced MDD of the example two-phase system.

Fig. 6. The system reliability calculation procedure with the upwards algorithm for the example two-phase system.
evaluation of the DB-PMS. Referring to the corresponding CPU times, it is clear that the complexity of the reduction procedure is comparable to the complexity of the construction procedure, while the time consumption of the upwards MDD evaluation algorithm is negligible.

We also applied the Monte Carlo simulation to validate the proposed method. As can be seen from Table 5 and Fig. 8, the system reliability results obtained using the proposed method are consistent with the simulation results. For the simulation method, the CPU time depends on the number of simulation runs, which can be approximately estimated by $5.4 \times 10^{-5}$ $Ns$ in our case, where $N$ is the number of runs. Though the simulation method seems to take less CPU time than the proposed method in some cases when the number of simulations is not very large, one cannot be confident with the exact system reliability due to the sampling variation. For example, we estimated the DB-PMS reliability for $D=(38,65,39,40)$ with $N=2 \times 10^4$ runs (the CPU time is about 10.8 s) and replicated this procedure for ten times. We found that the system reliability varies from 0.7296 to 0.7346. So if the system reliability design criterion is set to 0.73, it may happen that the simulated result cannot meet this criterion. However, the true value of the system reliability is 0.7315 in this case, which indicates that the simulation results may lead to improper design decisions. It is recognized that the simulation method is a
reduce the size of the MDD. A reduction procedure is put forward to further simplify the MDD and an upwards evaluation algorithm is proposed to calculate the system reliability based on the reduced MDD. Two illustrative examples are given to illustrate the applicability of the proposed method for the reliability evaluation of the DB-PMS with parallel structure. The reliability analysis of the DB-PMS with other structures, e.g., series-parallel DB-PMS, will be investigated in our future work.

6. Conclusion

PMS abound in the real-world applications such as aerospace, nuclear power plants, and distributed computing systems. Modeling and evaluation of PMS is much more complicated than single-phase systems due to the dependence among the phases as well as dynamics in system configuration, demands, and component behavior. In this paper we evaluate the reliability of a demand-based PMS with parallel structure subject to fault level coverage using the MDD-based combinatorial method. We first propose a straightforward construction procedure to construct the system MDD, which is applicable for analyzing the system subject to different mission demands. Hence, this approach is only suitable for small-scale systems. However, we propose an alternative construction procedure with a branching truncation mechanism which can reduce the size of the MDD. A reduction procedure is put forward to further simplify the MDD and an upwards evaluation algorithm is proposed to calculate the system reliability based on the reduced MDD. Two illustrative examples are given to illustrate the applicability of the proposed method for the reliability evaluation of the DB-PMS with parallel structure. The reliability analysis of the DB-PMS with other structures, e.g., series-parallel DB-PMS, will be investigated in our future work.

Table 6

<table>
<thead>
<tr>
<th>Demand for the single-phase system: D</th>
<th>BDD construction: NONSN (CPU time)</th>
<th>BDD reduction: NONSN (CPU time)</th>
<th>System reliability (upwards evaluation time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>650 (0.038497s)</td>
<td>82 (0.015554s)</td>
<td>0.9751 (0.000614s)</td>
</tr>
<tr>
<td>45</td>
<td>3918 (0.085396s)</td>
<td>141 (0.097610s)</td>
<td>0.9929 (0.000726s)</td>
</tr>
<tr>
<td>40</td>
<td>2406 (0.479442s)</td>
<td>219 (1.053205s)</td>
<td>0.9942 (0.000932s)</td>
</tr>
<tr>
<td>38</td>
<td>66700 (0.886223s)</td>
<td>245 (2.249223s)</td>
<td>0.9943 (0.001004s)</td>
</tr>
</tbody>
</table>

powerful technique in the system reliability estimation. However, the advantages of the proposed method over the simulation method lie in two-fold. First, the proposed method can produce exact system reliability results, instead of approximate results that vary in different simulations. Second, once the system MDD model is constructed, it can be reused and the system reliability can be calculated with the upwards algorithm efficiently. Specifically, the system reliability can be easily recalculated when the component failure probabilities are changed (for example, due to the change of mission time, the change of the component’s lifetime distribution or the change of acceleration factors). However, in the simulation method, if the failure probability of any component in any phase is changed, a full simulation has to be implemented to estimate the system reliability.

Another intuitive way to calculate the system reliability is by enumeration. However, in this case there are $5^{20} \approx 9.54 \times 10^{11}$ possible combinations for 20 components where each component may fail in any of four phases or may not fail in any phase. Only enumerating these possible combinations would take hundreds of thousands seconds for the 2.93 GHz CPU, so it is practically infeasible to resort to the exhaustive method for the analysis of DB-PMS.

To show the difference between the PMS and the single-phase system, we consider a single-phase system derived from this example case: if the system capacity meets the mission demand in first phase, the system is thought to be successful. The inputs for the single-phase system are assumed to be identical to those in the four-phase system. Then the MDD reduces to the BDD. Table 6 lists the NONSN in the BDD, the CPU time and the system reliability for the single-phase system subject to different demands. Compared with the four-phase system, apparently there are fewer non-sink nodes in the BDD for the single-phase system during the BDD construction and after the reduction. Since the system only needs to survive one phase, the system reliability increases a lot. Again, the CPU time consumed by the upwards algorithm is negligible compared with the construction and the reduction time.

References


