

Ji-Min LU
Mary Ann LUNDTEIGEN
Yiliu LIU
Xiao-Yue WU

FLEXIBLE TRUNCATION METHOD FOR THE RELIABILITY ASSESSMENT OF PHASED MISSION SYSTEMS WITH REPAIRABLE COMPONENTS

ZASTOSOWANIE METODY ELASTYCZNEGO OBCIĘCIA DO OCENY NIEZAWODNOŚCI SYSTEMÓW O ZADANIACH OKRESOWYCH Z ELEMENTAMI NAPRAWIALNYMI

Phased-mission systems (PMS) are the system in which the component stresses and the system configuration may change over time. Real-world PMS usually consist of a large number of repetitive phases and repairable components. Existing approaches for the reliability analysis of this kind of PMS tend to suffer from the problem of state explosion or binary-decision-diagram (BDD) explosion. This paper presents a truncation method based on the BDD and Markov chains to solve these scaling issues. In our approach, the truncation mitigates the BDD explosion and broadens the applicability of the BDD & Markov method. Different from the classic truncations, our truncation limit is flexible to ensure the truncation error is lower than the predefined threshold. The advantages of the proposed method are illustrated through two practical PMS which are challenging to classic non-simulation approaches.

Keywords: flexible truncation limit; phased mission systems; reliability evaluation; repairable components.

Systemy o zadaniach okresowych (phased mission systems, PMS) to takie systemy, w których naprężenia elementów składowych oraz konfiguracja systemu mogą z czasem ulegać zmianie. W warunkach rzeczywistych, PMS zazwyczaj charakteryzują się dużą liczbą powtarzalnych faz zadaniowych i składają się z wielu naprawialnych elementów. Istniejące metody analizy niezawodności tego typu systemów niestety posiadają ograniczenia związane z problemem eksplozji stanów lub eksplozji diagramów binarnych decyzji (binary decision diagram, BDD) Praca przedstawia metodę obcinania opartą na BDD oraz łańcuchach Markowa, która pozwala rozwiązać wspomniane problemy złożoności obliczeniowej. W proponowanym podejściu, obcięcie minimalizuje eksplozję BDD zwiększając możliwości zastosowania metody opartej na BDD oraz łańcuchach Markowa. W odróżnieniu od klasycznego obcinania, w opracowanej przez nas metodzie granica obciążenia jest elastyczna co pozwala zredukować błąd obciążenia poniżej wcześniej określonego progu. Zalety proponowanej metody zilustrowano na przykładzie dwóch stosowanych w praktyce systemów PMS, które stanowią wyzwanie dla klasycznych metod niesymulacyjnych.

Słowa kluczowe: elastyczna granica obciążenia; systemy o zadaniach okresowych; ocena niezawodności; elementy naprawialne.

1. Introduction

In many real-world applications such as airliner, railway, and satellite systems, operation of missions often involves multiple tasks or phases that must be accomplished in sequence. During each phase, some equipments may play different roles and subject to different environmental conditions. Thus, the system configuration, success criteria, and component behavior may change from phase to phase. This kind of systems are commonly termed as the phased-mission systems (PMS). A typical example of PMS is the monitoring system in the satellite-launching mission which involves the launching, separation, and orbiting phases. Some equipments, such as the ground-station radars, may suffer from increasing stresses in the separation phase and becomes idle in other phases. Because of the dependence problem, the reliability of PMS is not the product of reliabilities of individual phases. Hence, the reliability analysis of PMS is more complex than that of the single-phase systems.

Over the past four decades, substantial progress has been made in the reliability analysis of PMS [3]. Basically, existing methodologies

can be categorized into the simulation and the analytical methods. The main advantage of the simulation methods [6, 19, 25] is their wide applicability to a variety of scenarios, while the strength of the analytical approaches lies in the accuracy of algorithm results. The analytical approaches can be further classified into the state-based, the combinatorial, and the modular methods. Typical state-based methods, such as the Markovian models [1, 4, 28, 31], are commonly used to analyze the PMS with repairable components. A commonly known challenge facing the state-based methods is the state explosion problem when the number of components becomes large. On the other hand, the binary decision diagrams (BDD) based methods [2, 15, 16, 18, 22, 27, 29, 30] (belong to combinatorial methods) are efficient for the PMS with many components. However, BDD based methods can suffer a similar explosion problem (node or path explosion) when the number of phases becomes large. This problem is known as the BDD explosion in the PMS analysis.

The modular methods [13, 14, 21, 24, 25], which integrate the state-space and the combinatorial methods, are efficient in analyzing the PMS with a multitude of repairable components. A classic rep-

representative of the modular methods is the BDD & Markov approach [25] which is designed for the PMS with many exponentially distributed components. As the further work of the BDD & Markov method, Shrestha et al. [24] proposed the multistate multivalued decision diagram for the reliability analysis of PMS with multi-state repairable components. Another further work [13] is the component-behavior model which analyzes the PMS with combinatorial phase requirements and repairable components. However, the BDD & Markov method (and its further works) cannot avoid the BDD explosion problem because the cross-phase BDD has to be assessed from the first phase to the last phase.

In order to overcome the scaling issue, extensive research efforts have been expended in the analysis of approximations, bounds, and truncations. A common practice [5, 11, 12, 17, 19] is to apply truncations to BDD or cut sets, that is, to remove the BDD nodes (or BDD paths) whose probabilities are less than the truncation limit. However, a lot of researchers [5, 7, 12] find that a fixed truncation limit may be the major source of estimation error. In this paper, we implement truncations to reduce the computational complexity of the BDD & Markov method for the analysis of large PMS. Furthermore, our truncation limit will decrease as the truncation proceeds. This truncation strategy can keep the overall error under the user control, and avoids the complex discussion about error estimation in the literature.

The remainder of the paper is organized as follows. Section 2 presents the proposed method which integrates BDD, Markov chains, and truncations. Section 3 illustrates our approach through two real-world PMS. The efficiency of our approach is compared with the conventional BDD & Markov method and the Petri-net simulation. Lastly, Section 4 gives conclusions and future directions of our work.

2. Proposed method for large PMS analysis

2.1. Assumptions & method overview

Throughout the paper, we make the following assumptions for our approach. (1) The life and the repair time of components are independent variables of exponential distributions. (2) The mission is assumed to fail if the system fails in any phase. (3) Repaired components can be reused only in the next phase. (This assumption exists in the BDD & Markov method as well.)

Generally, our approach consists of mainly three steps, that is, Step 1 - Generate component-behavior vectors (CBV) based on BDD.

Step 2 - Compute path vectors (PV) phase by phase.

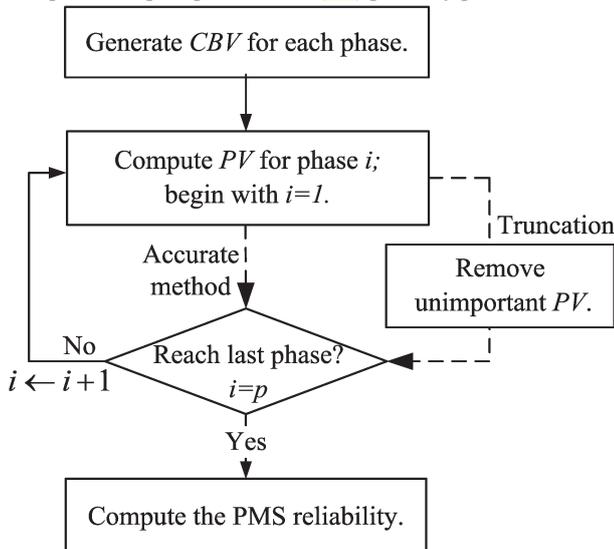


Fig. 1. Flowchart of the proposed method

Step 3 – Sum up the probabilities of the last-phase PV to obtain the PMS reliability.

In the following, we first propose our model without the truncation. When the number of phases becomes large, we need to use the truncation step to reduce the time and the space consumption of our model. The flowchart of our approach is given in Fig. 1.

2.2. Model without truncation

The concept of component-behavior vectors (CBV) was first proposed in [13]. CBVs are in essence equivalent to the BDD paths (readers may refer to [10, 23] for BDD basics). Consider the system whose reliability block diagram (RBD) is shown in Fig. 2. CBVs can be obtained through the enumeration of the paths which link the root node and the sink node \square in the BDD. In CBVs, the matrix $U_i^{(k)}$ corresponds to the “true” edge of node k , while $D_i^{(k)}$ corresponds to the “false” edge. $E_i^{(k)}$ is used if the BDD path does not contain the decision node of k . The expressions of $E_i^{(k)}$, $U_i^{(k)}$, and $D_i^{(k)}$ can be found in [13, 25]. For instance, $U_i^{(k)}$ (for binary-state component) is of the form:

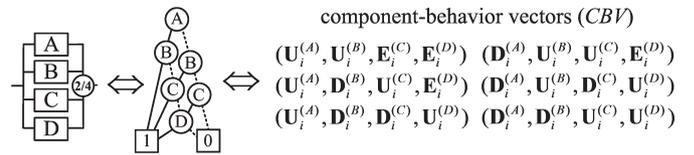


Fig. 2. Generation of CBVs

$$E_i^{(k)} = \exp \begin{pmatrix} -\lambda_i^{(k)} & \lambda_i^{(k)} \\ \mu_i^{(k)} & -\mu_i^{(k)} \end{pmatrix} \cdot T_i \quad (1)$$

where T_i is the duration of phase i . $\lambda_i^{(k)}$ and $\mu_i^{(k)}$ is the failure rates and the repair rates of k in phase i , respectively.

After we generate CBVs for each phase, the reliability of PMS can be assessed through path vectors PV_i . PV_i is defined as:

$$PV_i = \begin{cases} CBV_i, & i=1 \\ PV_{i-1} \circ CBV_i, & i>1 \end{cases} \quad (2)$$

where i is the phase index. The operator “ \circ ” represents the element-by-element multiplication, which is also known as the Hadamard product (or the entrywise product) [9]. Take the system in Fig. 3 for instance, there are two CBVs in phase 1 and three CBVs in phase 2. They are:

$$CBV_1^{(1)} = (U_1^{(A)}, E_1^{(B)}, E_1^{(C)}, E_1^{(D)}) \quad (3)$$

$$CBV_1^{(2)} = (D_1^{(A)}, U_1^{(B)}, E_1^{(C)}, E_1^{(D)})$$

$$CBV_2^{(1)} = (U_2^{(A)}, E_2^{(B)}, E_2^{(C)}, E_2^{(D)})$$

$$CBV_2^{(2)} = (D_2^{(A)}, E_2^{(B)}, U_2^{(C)}, E_2^{(D)}) \quad (4)$$

$$CBV_2^{(3)} = (D_2^{(A)}, E_2^{(B)}, D_2^{(C)}, U_2^{(D)})$$

According to the definition of PVs , PVs of phase 1 are identical to $CBVs$, i.e.,

$$PV_1^{(q)} = CBV_1^{(q)} \quad (q = 1, 2, \dots) \quad (5)$$

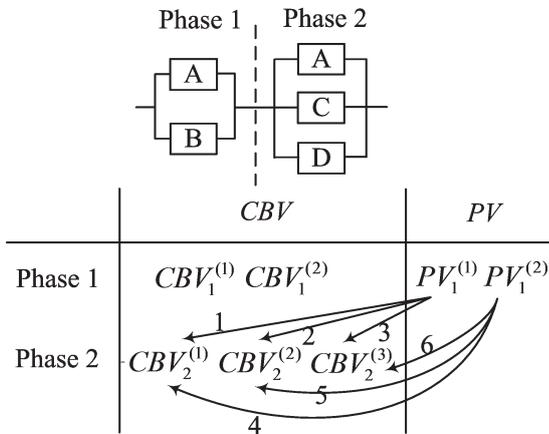


Fig. 3. Generation of PVs

PVs of phase 2 are the entrywise product of PV_1 and CBV_2 . For instance, $PV_2^{(5)}$ (fifth path vector in phase 2) can be expressed as:

$$PV_2^{(5)} = PV_1^{(2)} \circ CBV_2^{(2)} = CBV_1^{(2)} \circ CBV_2^{(2)} = (\mathbf{D}_1^{(A)} \mathbf{D}_2^{(A)}, \mathbf{U}_1^{(B)} \mathbf{E}_2^{(B)}, \mathbf{E}_1^{(C)} \mathbf{U}_2^{(C)}, \mathbf{E}_1^{(D)} \mathbf{E}_2^{(D)}) \quad (6)$$

When the final-phase PVs are generated, the PMS reliability is the sum of probabilities of the last-phase PVs , i.e.,

$$R_{PMS} = \sum_{j=1}^p \Pr\{PV_p^{(j)}\} \quad (7)$$

where p is the index of the last phase. $\Pr\{PV_p^{(j)}\}$ is the probability of $PV_p^{(j)}$. Suppose $PV_p^{(j)} = (\mathbf{a}_1, \dots, \mathbf{a}_n)$, $\Pr\{PV_p^{(j)}\}$ is defined as:

$$\Pr\{PV_p^{(j)}\} = (\mathbf{v}_0^{(1)} \cdot \mathbf{a}_1 \cdot \mathbf{1}') \cdot (\mathbf{v}_0^{(2)} \cdot \mathbf{a}_2 \cdot \mathbf{1}') \cdot \dots \cdot (\mathbf{v}_0^{(n)} \cdot \mathbf{a}_n \cdot \mathbf{1}') \quad (8)$$

where $\mathbf{v}_0^{(k)} = (1, 0)$ if the component k is initially operational. The column vector $\mathbf{1}' = (1, 1)$ is used to transfer $\mathbf{v}_0^{(k)} \cdot \mathbf{a}_k$ into the scalar. For instance, the probability of $PV_2^{(5)}$ is given by:

$$\begin{aligned} \Pr\{PV_2^{(5)}\} &= \Pr\{(\mathbf{D}_1^{(A)} \mathbf{D}_2^{(A)}, \mathbf{U}_1^{(B)} \mathbf{E}_2^{(B)}, \mathbf{E}_1^{(C)} \mathbf{U}_2^{(C)}, \mathbf{E}_1^{(D)} \mathbf{E}_2^{(D)})\} \\ &= (\mathbf{v}_0^{(A)} \mathbf{D}_1^{(A)} \mathbf{D}_2^{(A)} \mathbf{1}') \cdot (\mathbf{v}_0^{(B)} \mathbf{U}_1^{(B)} \mathbf{E}_2^{(B)} \mathbf{1}') \cdot \\ &\quad (\mathbf{v}_0^{(C)} \mathbf{E}_1^{(C)} \mathbf{U}_2^{(C)} \mathbf{1}') \cdot (\mathbf{v}_0^{(D)} \mathbf{E}_1^{(D)} \mathbf{E}_2^{(D)} \mathbf{1}') \end{aligned}$$

Suppose the PMS contains p phases, we can calculate the PMS reliability **using** after we obtain all $\Pr\{PV_p^{(j)}\}$. The proposed algorithm (without truncations) can be summarized by Fig. 4.

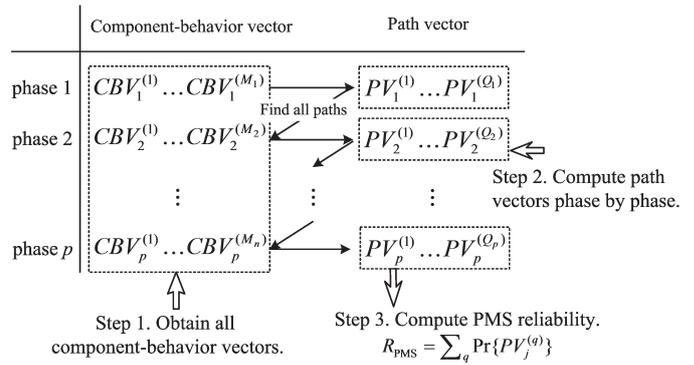


Fig. 4. Proposed algorithm without truncation

2.3. Truncation with flexible threshold

Many real-world PMS may contain thousands of phases which may lead to enormous space consumption in the above algorithm. On this occasion, the truncation step is necessary to reduce the computational complexity. Consider the Fig. 5 PMS which contains several repetitive phases. It is inefficient to calculate all $\Pr\{PV_i\}$ because the number of PVs increases exponentially with the phase index i . In the BDD & Markov approach [13, 24, 25], a similar problem exists because the top-down algorithm can lead to the BDD path explosion (or bottom-up algorithm leads to BDD node explosion) when the number of phases becomes large.

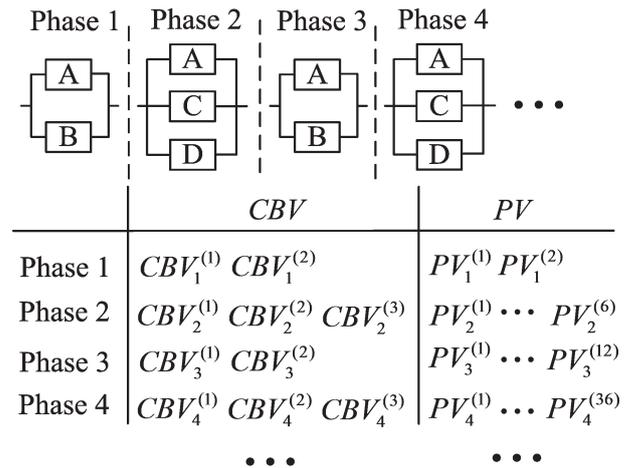


Fig. 5. Exponential increase in the number of PVs

In order to solve the scaling problem, we can use the classical truncation methods [5, 11, 12, 17, 19] with fixed truncation limit (i.e. fixed threshold). However, the error analysis associated with a fixed truncation limit may be very complex. In addition, some works [5, 7] show that the truncation error is highly sensitive to the component parameters. Here, we present a **flexible** truncation limit to guarantee that the truncation error is less than the predefined "maximum permissible error" (**MPE**). The flexible truncation limit γ_i of phase i is defined as:

$$\gamma_i = \frac{MPE - \Pr\{PV\}}{Num(PV_i)} \quad (9)$$

where $Num(PV_i)$ is the number of PVs which are removed during phase i . With γ_i , we remove PV_i if $\Pr\{PV_i\} < \gamma_i$. Every time we de-

lete a path vector, the value of γ_i will decrease. MPE is a predefined algorithm parameter.

In the following, we demonstrate that the total truncation error is less than (or equals to) MPE . Let PV_i^Δ to be the path vector removed when the algorithm reaches phase i . By removing PV_i^Δ , we also eliminate some PVs in the later phase. Take Fig. 6 for example, suppose $PV_2^{(1)}$ (first PV in phase 2) is removed during phase 2. PVs stem from $PV_2^{(1)}$ are also removed (including $PV_3^{(1)}$, $PV_3^{(2)}$, and $PV_4^{(1)} \sim PV_4^{(6)}$). Since $\sum_i \Pr\{CBV_i\} < 1$, we can see that $\Pr\{PV_i^\Delta\}$ is bigger than (or equals to) the sum of probabilities of PVs which stem from PV_i^Δ , i.e.,

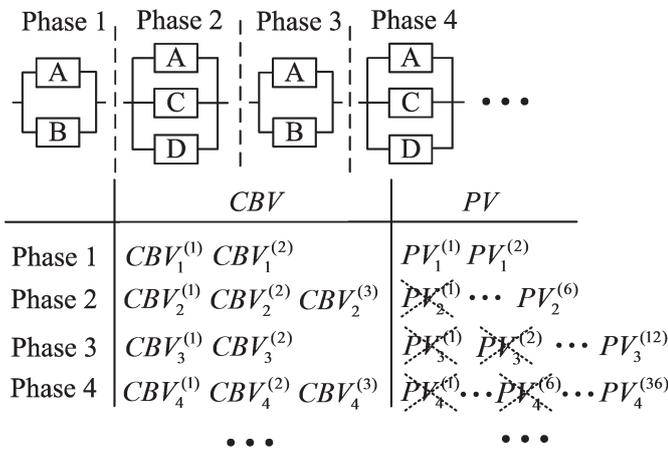


Fig. 6. Truncation of PVs

$$\Pr\{PV_i^\Delta\} \geq \sum \Pr\{PV_{i+1}^*\}; (PV_{i+1}^* \text{ stems from } PV_i^\Delta) \quad (10)$$

By repeating have:

$$\Pr\{PV_i^\Delta\} \geq \sum \Pr\{PV_{i+1}^*\} \geq \dots \geq \sum_s \Pr\{PV_q^{(s)}\} \quad (11)$$

where $PV_q^{(s)}$ is the last-phase PVs which stem from PV_i^Δ . When the algorithm reaches the last phase, we sum up the probabilities of all removed PV_i^Δ , and then:

$$MPE \geq \sum_i \Pr\{PV_i^\Delta\} \geq \sum_i \sum_s \Pr\{PV_q^{(s)}\} = |R_{PMS} - R'_{PMS}| = error \quad (12)$$

where $MPE \geq \sum_i \Pr\{PV_i^\Delta\}$ holds because of the definition of γ_i .

The inequality (12) shows that the total truncation error is less than the user-defined MPE . Overall, the proposed method with truncations can be summarized by Fig. 7.

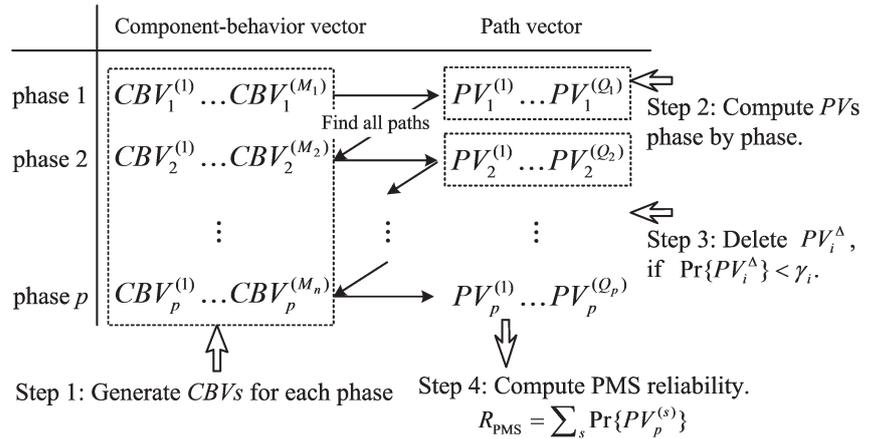


Fig. 7. Proposed algorithm with truncations

2.4. Determination of MPE

From above algorithm, we find that a smaller MPE parameter leads to a more accurate reliability result, but at the cost of more memory and time expense. Generally, MPE is determined by the following steps:

Step 1 – Set $MPE=0$ and check whether the algorithm is able to generate a reliability result. Empirically, if the algorithm cannot generate a result within 3 min, it suggests that the memory space is not sufficient (computing with virtual memory is unacceptably time-consuming). In this case, MPE should increase (see Step 2).

Step 2 – Let $MPE=10^{-9}$, and run the algorithm to see the algorithm performance. If the algorithm still encounter the problem of insufficient memory, move to Step 3.

Step 3 – Increase MPE by 10 times, and run the truncation algorithm with new MPE . Repete Step 3 until the truncation algorithm is able to generate a result. The appropriate MPE is the first value which guarantees the algorithm can generates a result.

For most PMS cases, our truncation algorithm is able to generate a result with $MPE=0.1$. Above 3-step procedure can be summarized as Fig. 8. In summary, MPE is determined by the case concerned and the performance of user's computer. We need to test different MPE sample to determine the appropriate MPE .

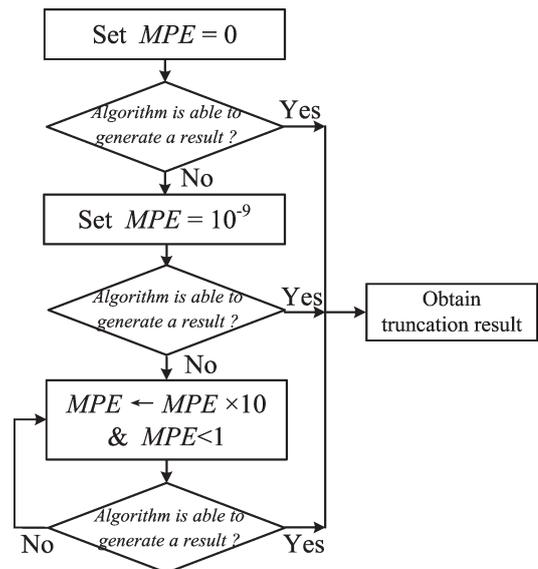


Fig. 8. Procedure to determine maximum permissible error

3. Case study

3.1. Train-speed monitoring mission

The surveillance of the train speed is an important mission in railway administration. The speed of the train can be measured by the sensors installed near railway stations, or by the radars installed along the railway. Consider a train which travels across three cities for two laps in a day, as shown in Fig. 9. The monitoring system is operational

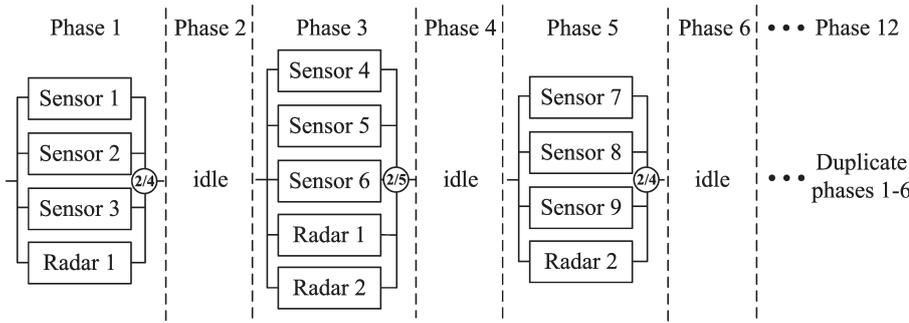


Fig. 9. Train-speed monitoring mission

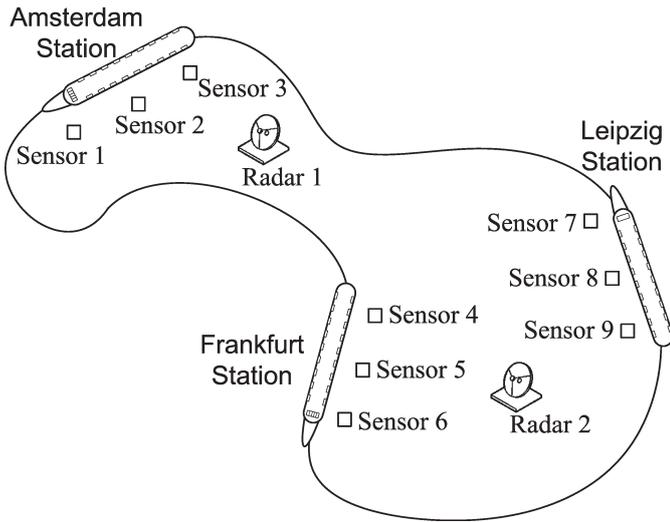


Fig. 10. PMS model for the speed-monitoring mission

Table 1. Parameters of the speed-monitoring PMS

Parameters	Value (Time unit: hour)
Phase duration T_i (i - phase index)	$T_i = \begin{cases} 0.2, & i = 1,3,5,\dots,17 \\ 3, & i = 2,4,6,\dots,18 \end{cases}$
Failure rates	$\lambda_K = \begin{cases} 2 \cdot 10^{-3}, & K = \text{Sensors} \\ 10^{-3}, & K = \text{Radars} \end{cases}$ (λ_K remains fixed in all phases.)
Repair rates μ_K	$\mu_K = 0.01$ (for all components) (μ_K remains fixed in all phases.)

as long as two sets of data can be simultaneously recorded. This system can be modeled as the PMS whose RBD is shown in Fig. 10.

Suppose all equipments in Fig. 10 are repairable, there are three kinds of methods available for the reliability analysis of this PMS. Apparently, the conventional Markovian method [1] is not efficient because of the enumeration of 2^{11} states. The BDD & Markov method [13, 24, 25] and our approach can generate the results using less space than the Markovian approach. In comparison with the BDD & Markov method, the main advantage of our approach is the truncation step which significantly reduces the computation time (see the experimen-

tal results in Table 2). In this PMS, we assume the equipments may fail even if they are idle. The system parameters are presented in Table 1.

In our approach, we first generate **CBVs** of phases 1-6 (see Fig. 11). **CBVs** of phases 7-12 are the same as that of phases 1-6. Secondly, we calculate **PVs** phase by phase using. For instance, $PV_3^{(1)}$ is of the form:

$$PV_3^{(1)} = (\mathbf{U}_1^{(S1)} \mathbf{E}_2^{(S1)} \mathbf{E}_3^{(S1)}, \mathbf{E}_1^{(S4)} \mathbf{E}_2^{(S4)} \mathbf{U}_3^{(S4)}) \quad (13)$$

Finally, given $MPE=10^{-6}$, the PMS reliability is given by

$$R_{PMS}(\sum_{j=1}^{18} T_j) = \sum_s \Pr\{PV_{18}^{(s)}\} \quad (14)$$

	CBV	PV
Phase 1	$CBV_1^{(1)} = (\mathbf{U}_1^{(S1)}, \mathbf{U}_1^{(S2)}, \mathbf{E}_1^{(S3)}, \mathbf{E}_1^{(R1)})$ $CBV_1^{(2)} = (\mathbf{U}_1^{(S1)}, \mathbf{D}_1^{(S2)}, \mathbf{U}_1^{(S3)}, \mathbf{E}_1^{(R1)})$ $CBV_1^{(3)} = (\mathbf{U}_1^{(S1)}, \mathbf{D}_1^{(S2)}, \mathbf{D}_1^{(S3)}, \mathbf{U}_1^{(R1)})$ $CBV_1^{(4)} = (\mathbf{D}_1^{(S1)}, \mathbf{U}_1^{(S2)}, \mathbf{U}_1^{(S3)}, \mathbf{E}_1^{(R1)})$ $CBV_1^{(5)} = (\mathbf{D}_1^{(S1)}, \mathbf{U}_1^{(S2)}, \mathbf{D}_1^{(S3)}, \mathbf{U}_1^{(R1)})$ $CBV_1^{(6)} = (\mathbf{D}_1^{(S1)}, \mathbf{D}_1^{(S2)}, \mathbf{U}_1^{(S3)}, \mathbf{U}_1^{(R1)})$	$CBV_1^{(1)} \sim CBV_1^{(6)}$
Phase 3	$CBV_3^{(1)} = (\mathbf{U}_3^{(S4)}, \mathbf{U}_3^{(S5)}, \mathbf{E}_3^{(S6)}, \mathbf{E}_3^{(R1)}, \mathbf{E}_3^{(R2)})$ $CBV_3^{(2)} = (\mathbf{U}_3^{(S4)}, \mathbf{D}_3^{(S5)}, \mathbf{U}_3^{(S6)}, \mathbf{E}_3^{(R1)}, \mathbf{E}_3^{(R2)})$ $CBV_3^{(3)} = (\mathbf{U}_3^{(S4)}, \mathbf{D}_3^{(S5)}, \mathbf{D}_3^{(S6)}, \mathbf{U}_3^{(R1)}, \mathbf{E}_3^{(R2)})$ $CBV_3^{(4)} = (\mathbf{U}_3^{(S4)}, \mathbf{D}_3^{(S5)}, \mathbf{D}_3^{(S6)}, \mathbf{D}_3^{(R1)}, \mathbf{U}_3^{(R2)})$ $CBV_3^{(5)} = (\mathbf{D}_3^{(S4)}, \mathbf{U}_3^{(S5)}, \mathbf{U}_3^{(S6)}, \mathbf{E}_3^{(R1)}, \mathbf{E}_3^{(R2)})$ $CBV_3^{(6)} = (\mathbf{D}_3^{(S4)}, \mathbf{U}_3^{(S5)}, \mathbf{D}_3^{(S6)}, \mathbf{U}_3^{(R1)}, \mathbf{E}_3^{(R2)})$ $CBV_3^{(7)} = (\mathbf{D}_3^{(S4)}, \mathbf{U}_3^{(S5)}, \mathbf{D}_3^{(S6)}, \mathbf{D}_3^{(R1)}, \mathbf{U}_3^{(R2)})$ $CBV_3^{(8)} = (\mathbf{D}_3^{(S4)}, \mathbf{D}_3^{(S5)}, \mathbf{U}_3^{(S6)}, \mathbf{U}_3^{(R1)}, \mathbf{E}_3^{(R2)})$ $CBV_3^{(9)} = (\mathbf{D}_3^{(S4)}, \mathbf{D}_3^{(S5)}, \mathbf{U}_3^{(S6)}, \mathbf{D}_3^{(R1)}, \mathbf{U}_3^{(R2)})$ $CBV_3^{(10)} = (\mathbf{D}_3^{(S4)}, \mathbf{D}_3^{(S5)}, \mathbf{D}_3^{(S6)}, \mathbf{U}_3^{(R1)}, \mathbf{U}_3^{(R2)})$	$PV_3^{(1)} \sim PV_3^{(60)}$
Phase 5	$CBV_5^{(1)} = (\mathbf{U}_5^{(S7)}, \mathbf{U}_5^{(S8)}, \mathbf{E}_5^{(S9)}, \mathbf{E}_5^{(R2)})$ $CBV_5^{(2)} = (\mathbf{U}_5^{(S7)}, \mathbf{D}_5^{(S8)}, \mathbf{U}_5^{(S9)}, \mathbf{E}_5^{(R2)})$ $CBV_5^{(3)} = (\mathbf{U}_5^{(S7)}, \mathbf{D}_5^{(S8)}, \mathbf{D}_5^{(S9)}, \mathbf{U}_5^{(R2)})$ $CBV_5^{(4)} = (\mathbf{D}_5^{(S7)}, \mathbf{U}_5^{(S8)}, \mathbf{U}_5^{(S9)}, \mathbf{E}_5^{(R2)})$ $CBV_5^{(5)} = (\mathbf{D}_5^{(S7)}, \mathbf{U}_5^{(S8)}, \mathbf{D}_5^{(S9)}, \mathbf{U}_5^{(R2)})$ $CBV_5^{(6)} = (\mathbf{D}_5^{(S7)}, \mathbf{D}_5^{(S8)}, \mathbf{U}_5^{(S9)}, \mathbf{U}_5^{(R2)})$	$PV_5^{(1)} \sim PV_5^{(360)}$
Phase 2, 4, 6	$CBV_i^{(1)} = (\mathbf{E}_i^{(S1)}, \dots, \mathbf{E}_i^{(R2)})$ ($i = 2, 4, 6$)	

Fig. 11. CBVs of phase 1-6

Table 2. Reliability and computation time for the speed-monitoring PMS

Time (Hour)	Traditional BDD & Markov		Proposed method (MPE=10 ⁻⁶)		Petri-net simulation
	PMS reliability	Computation time (s)	PMS reliability	Computation time (s)	PMS reliability
9.6 (phase-6 end)	0.999995	0.07	0.999995	0.04	0.999995
19.2 (phase-12 end)	0.999919	23.9	0.999918	0.93	0.999920

In order to verify the correctness of our algorithm, we compare the non-simulation approaches with the Petri-net simulation. From Table 2, we can see that three kinds of methods generate the similar reliabilities for this PMS. The non-simulation algorithms are programmed by the MATLAB language, and implemented with a 1.8 GHz processor. The average computation time in Table 2 is recorded without multi-threading. And we can see that our approach is much less time consuming than the BDD & Markov method. In the Petri-net simulation, the model contains one Petri net for the phase-index increment (with low priority), one Petri net for the detection of system failure, and 11 Petri nets for the independent equipments. We carry out the Petri-net simulation using the GRIF software [8] with 10⁷ iterations. In the next instance, we show that the truncation step is indispensable in avoiding the excessive memory cost of the traditional BDD & Markov method.

3.2. Regular test mission in the oil and gas system

In the oil and gas industry, the regular test is one essential part of safety procedures. Consider a pressure protection system which is used to stop flow in case of overpressure. Equipments in the system may suffer from higher pressure during the test phases than the normal phases. Hence, the pressure protection system can be model as the PMS whose structure remains fixed during the entire mission, as shown in Fig. 12.

It can be seen from Fig. 12 that the phase 1 and the phase 2 are duplicated as the test is periodically carried out. In this PMS, components can be modeled as repairable if they can be quickly renewed after failure. Generally, the duration of the test phase is much shorter than that of the normal phases. Here, we suppose the life and

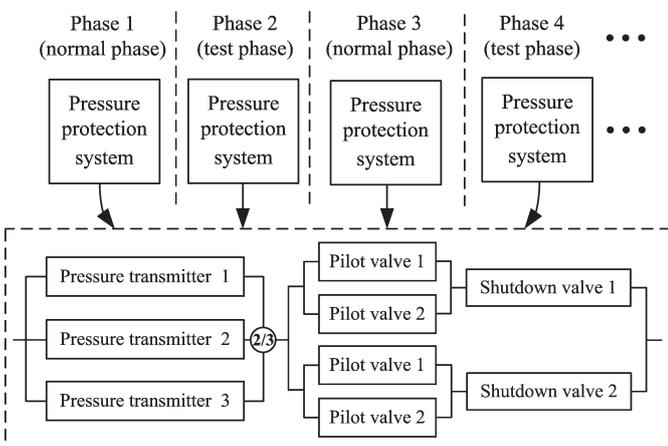


Fig. 12. PMS model for the pressure protection system

the repair time are exponentially distributed. Relevant parameters of the PMS are presented in Table 3.

In the traditional BDD & Markov method, the time cost and the space cost increases exponentially with the number of phases in the PMS. The experimental results in Table 4 show that the computing capacity of the BDD & Markov algorithm is 5 phases. Further computation will result in the problem of the insufficient compu-

Table 3. Parameters of the pressure-protection PMS

Parameters	Value (Time unit: hour)
Phase duration T_i	$T_i = \begin{cases} 168, & \text{normal phase.} \\ 2, & \text{test phase.} \end{cases}$
Failure rates λ_K	$\lambda_K = \begin{cases} 10^{-4}, & \text{normal phase.} \\ 5 \cdot 10^{-4}, & \text{test phase.} \end{cases}$ (for all components)
Repair rates μ_K	$\mu_K = 0.05$ (for all components)

Table 4. Reliability and computation time for the pressure-protection PMS

Time (Hour)	Traditional BDD & Markov		Proposed method (MPE=10 ⁻⁵)		Petri-net simulation
	PMS reliability	Computation time (s)	PMS reliability	Computation time (s)	PMS reliability
170 (end of phase 2)	0.998871	0.02	0.998871	0.02	0.998867
340 (end of phase 4)	0.997389	7.23	0.997388	1.22	0.997468
510 (end of phase 5)	0.995930	146	0.995927	3.73	0.996095
680 (end of phase 8)	Insufficient memory		0.994425	185	0.994594

ter memory. However, when we use the proposed flexible truncation (with MPE=10⁻⁵), the computing capacity of our approach increases to 8 phases. The computing capacity will increase if MPE gets bigger. In our experiment, the algorithms are programmed without multi-threading, and run on a 1.8 GHz processor to record the computation time.

The proposed method is validated by the Petri-net simulation using the GRIF software [8] with 10⁷ iterations. In the Petri-net model, there is one Petri-net determining the phase transition, one Petri net determining the PMS failure, and 9 Petri nets determining the states of 9 components. By comparing the BDD & Markov method with our approach, we can see that the truncation step is indispensable to obtain the PMS reliability. In many real-world applications, the PMS model often contains a large number of repetitive phases, and the components in the PMS are usually considered as repairable. The proposed method is efficient to analyze this kind of PMS.

4. Conclusion

In the modeling of practical PMS, the system normally consists of several subsystems, and the subsystems may comprise many equipments. To keep the PMS model concise, reliability engineers usually

neglect the details of equipment configurations, and consider each subsystem as an entity whose behavior is exponentially distributed. Hence, the number of components in the PMS may not be very large if the engineers want a small-scale model. However, most PMS models cannot avoid a large number of phases because the real-world mission usually contains many repetitive tasks. This paper proposes an efficient approach for the reliability assessment of these PMS.

When the PMS model contains a multitude of phases, existing approaches may suffer from the BDD-explosion problem. Our approach uses truncations to analyze the PMS with many phases, and uses BDD to analyze the PMS with many components. In our approach, the flexible truncation limit can keep the truncation error within the user-defined maximum permissible error. The case study shows that

the truncation is a necessary step to solve the explosion problem as the number of phases increases. A possible direction of future works is to explore the application of our method in the field of the aerospace, assembling, and nuclear power industry, and to explore better truncation strategies using classic methods.

Acknowledgement

This research is partly carried out with the Reliability, Availability, Maintainability and Safety (RAMS) group at the Norwegian University of Science and Technology (NTNU), and it is funded by the National Natural Science Foundation of China with the grant no. 71071159.

References

1. Alam M, Alsaggaf UM. Quantitative reliability evaluation of repairable phased-mission systems using Markov approach. *IEEE Transactions on Reliability* 1986; 35(5): 498-503, <http://dx.doi.org/10.1109/TR.1986.4335529>.
2. Amari SV, Xing LD, Shrestha A et al. Performability analysis of multistate computing systems using multivalued decision diagrams. *IEEE Transactions on Computers*, 2010; 59(10): 1419-1433, <http://dx.doi.org/10.1109/TC.2009.184>.
3. Burdick G, Fussell J, Rasmuson D et al. Phased mission analysis: A review of new developments and an application. *IEEE Transactions on Reliability* 1977; 26(1): 43-49, <http://dx.doi.org/10.1109/TR.1977.5215072>.
4. Cekyay B, Ozekici S. Performance measures for systems with Markovian missions and aging. *IEEE Transactions on Reliability* 2012; 61(3): 769-778, <http://dx.doi.org/10.1109/TR.2012.2207529>.
5. Ćepin M. Analysis of truncation limit in probabilistic safety assessment. *Reliability Engineering and System Safety* 2005; 87(3): 395-403, <http://dx.doi.org/10.1016/j.res.2004.06.009>.
6. Chew SP, Dunnett SJ, Andrews JD. Phased mission modelling of systems with maintenance-free operating periods using simulated Petri nets. *Reliability Engineering and System Safety* 2008; 93(7): 980-994, <http://dx.doi.org/10.1016/j.res.2007.06.001>.
7. Epstein S, Rauzy A. Can we trust PRA? *Reliability Engineering and System Safety* 2005; 88(3): 195-205, <http://dx.doi.org/10.1016/j.res.2004.07.013>.
8. Grif-workshop software, SATODEV LLC. 2014. <http://grif-workshop.com/>
9. Horn RA, Johnson C. *The Hadamard Product - Chapter 5 - Topics in matrix analysis*. London: Cambridge University Press, 1990
10. Huang HZ, Zhang H, Li Y. A new ordering method of basic events in fault tree analysis. *Quality and Reliability Engineering International* 2012; 28(3): 297-305, <http://dx.doi.org/10.1002/qre.1245>.
11. Jung WS, Han SH, Yang JE. Fast BDD truncation method for efficient top event probability calculation. *Nuclear Engineering and Technology* 2008; 40: 571-580, <http://dx.doi.org/10.5516/NET.2008.40.7.571>.
12. Jung WS, Yang J-E, Ha J. Development of measures to estimate truncation error in fault tree analysis. *Reliability Engineering and System Safety* 2005; 90(1): 30-6, <http://dx.doi.org/10.1016/j.res.2004.09.007>.
13. Lu J-M, Wu X-Y. Reliability evaluation of generalized phased-mission systems with repairable components. *Reliability Engineering and System Safety* 2014; 121: 136-145, <http://dx.doi.org/10.1016/j.res.2013.08.005>.
14. Lu J-M, Wu X-Y, Liu Yiliu, Mary Ann Lundteigen. Reliability analysis of large phased-mission systems with repairable components based on success-state sampling. *Reliability Engineering and System Safety* 2015; 142: 123-133, <http://dx.doi.org/10.1016/j.res.2015.05.010>.
15. Mo YC. New insights into the BDD-based reliability analysis of phased-mission systems. *IEEE Transactions on Reliability* 2009; 58(4): 667-678, <http://dx.doi.org/10.1109/TR.2009.2026804>.
16. Mo YC. Variable ordering to improve BDD analysis of phased-mission systems with multimode failures. *IEEE Transactions on Reliability*, 2009; 58(1): 53-57, <http://dx.doi.org/10.1109/TR.2008.2011673>.
17. Mo YC, Han J, Zhang Z et al. Approximate reliability evaluation of large-scale distributed systems. *Journal of Information Science and Engineering* 2014; 30(1): 25-41.
18. Mo YC, Xing L, Amari SV. A multiple-valued decision diagram based method for efficient reliability analysis of non-repairable phased-mission systems. *IEEE Transactions on Reliability* 2014; 63(1): 320-330, <http://dx.doi.org/10.1109/TR.2014.2299497>.
19. Mo YC, Zhong F, Zhao X. New results to BDD truncation method for efficient top event probability calculation. *Nuclear Engineering and technology* 2012; 44: 755-766, <http://dx.doi.org/10.5516/NET.03.2011.058>.
20. Mura I, Bondavalli A. Markov regenerative stochastic Petri nets to model and evaluate phased mission systems dependability. *IEEE Transactions on Computers* 2001; 50(12): 1337-1351, <http://dx.doi.org/10.1109/TC.2001.970572>.
21. Ou Y, Dugan JB. Modular solution of dynamic multi-phase systems. *IEEE Transactions on Reliability* 2004; 53(4): 499-508, <http://dx.doi.org/10.1109/TR.2004.837305>.
22. Peng R, Zhai QQ, Xing LD et al. Reliability of demand-based phased-mission systems subject to fault level coverage. *Reliability Engineering and System Safety* 2014; 121: 18-25, <http://dx.doi.org/10.1016/j.res.2013.07.013>.
23. Rauzy A. *Binary decision diagrams for reliability studies*. Handbook of performability engineering 381-396. Berlin: Springer Press, 2008, http://dx.doi.org/10.1007/978-1-84800-131-2_25.
24. Shrestha A, Xing LD, Dai YS. Reliability analysis of multistate phased-mission systems with unordered and ordered states. *IEEE Transactions on Systems Man and Cybernetics Part a - Systems and Humans* 2011; 41(4): 625-636, <http://dx.doi.org/10.1109/TSMCA.2010.2089513>.
25. Wang D, Trivedi KS. Reliability analysis of phased-mission system with independent component repairs. *IEEE Transactions on Reliability* 2007; 56(3): 540-551, <http://dx.doi.org/10.1109/TR.2007.903268>.

26. Wu Xin-Yang, Wu X-Y. Extended object-oriented Petri net model for mission reliability simulation of repairable pms with common cause failures. *Reliability Engineering and System Safety* 2015; 136: 109-119, <http://dx.doi.org/10.1016/j.res.2014.11.012>.
27. Xing L. Reliability evaluation of phased-mission systems with imperfect fault coverage and common-cause failures. *IEEE Transactions on Reliability* 2007; 56(1): 58-68, <http://dx.doi.org/10.1109/TR.2006.890900>.
28. Xing L, Dugan JB, Morrisette BA. Efficient reliability analysis of systems with functional dependence loops. *Eksploatacja i Niezawodność – Maintenance and Reliability* 2009; 43(3): 65-69.
29. Xing L, Levitin G. Bdd-based reliability evaluation of phased-mission systems with internal/external common-cause failures. *Reliability Engineering and System Safety* 2013; 112: 145-153, <http://dx.doi.org/10.1016/j.res.2012.12.003>.
30. Zang X, Sun H, Trivedi KS. A BDD-based algorithm for reliability analysis of phased-mission systems. *IEEE Transactions on Reliability* 1999; 48(1): 50-60, <http://dx.doi.org/10.1109/24.765927>.
31. Zhang T, Bai GH, Guo B. Success probability model of phased mission systems with limited spares. *Eksploatacja i Niezawodność – Maintenance and Reliability* 2012; 14(1): 24-32.

Ji-Min LU

College of Information System and Management
National University of Defense Technology
De Ya Road 109, Changsha, Hunan 410073, China

Mary Ann LUNDTEIGEN**Yiliu LIU**

Department of Production and Quality Engineering
Norwegian University of Science and Technology, Valgrinda,
N-7491 Trondheim, Norway

Xiao-Yue WU

College of Information System and Management
National University of Defense Technology
De Ya Road 109, Changsha, Hunan 410073, China

E-mail: jeemanlv@gmail.com, mary.a.lundteigen@ntnu.no,
yiliu.liu@ntnu.no, xiaoyuewucn@gmail.com
