Using Reliability Analysis to Support Decision Making in Phased Mission Systems

Yang Zhang[1], Darren Prescott[1,2]

[1] Resilience Engineering Research Group, University of Nottingham, Nottingham, UK

[2] Corresponding Author: Darren.Prescott@nottingham.ac.uk

Abstract

Due to the environments in which they will operate, future autonomous systems must be capable of reconfiguring quickly and safely following faults or environmental changes. Past research has shown how, by considering autonomous systems to perform phased missions, reliability analysis can support decision making by allowing comparison of the probability of success of different missions following reconfiguration. Binary Decision Diagrams (BDDs) offer fast, accurate reliability analysis that could contribute to real-time decision making. However, phased mission analysis using existing BDD models is too slow to contribute to the instant decisions needed in time-critical situations. This paper investigates two aspects of BDD models that affect analysis speed: variable ordering and quantification efficiency. Variable ordering affects BDD size, which directly affects analysis speed. Here, a new ordering scheme is proposed for use in the context of a decision making process. Variables are ordered before a mission and reordering is unnecessary no matter how the mission configuration changes. Three BDD models are proposed to address the efficiency and accuracy of existing models. The advantages of the developed ordering scheme and BDD models are demonstrated in the context of their application within a reliability analysis methodology used to support decision making in an Unmanned Aerial Vehicle.

Key words: Decision Support, Reliability analysis, Binary Decision Diagrams, Variable ordering, Phased mission.

1 Introduction

Systems that perform a sequence of tasks in order to achieve a specific objective are called phased mission systems (PMS). The periods in which each of these successive tasks takes place are known as phases and the series of phases as a whole is known as a mission. A mission configuration is defined according to the tasks that must be completed, the time duration of each task and the sequence of the tasks. A mission is successfully completed if all of its phases are successfully completed. PMS are common in autonomous systems such as unmanned aerial vehicles (UAV), satellites and MARS rovers and autonomous transportation. In practice, PMS such as those carried out by a UAV are non-repairable since repairs are not possible during a mission.

The analysis of PMS is made complex by the dependencies that arise when considering the phases in which components fail, since component failures could lead to loss of functionality immediately or in a later phase, when the use of the system changes. Consideration of mutually exclusive component failure modes also introduces dependencies into the analysis, again making it more complex than the analysis of systems operating for single phases, which contain components that fail in single failure modes[1].

Past research has proposed the used of reliability analysis to support decision making for PMS[2]. Mission reliability was proposed as a key decision variable to determine whether or not a mission should continue in its present format. The reliability of a phased mission is defined as the probability that all phases in the mission are completed without failure.

The initial unreliability, calculated before the mission begins, is used to help determine whether the mission should start. An updated unreliability is calculated once the mission is underway, whenever new information is obtained about the system. If the updated unreliability is unacceptably high, then other mission configurations must be considered to eliminate potential risks.

In time-critical applications where alternative missions must be evaluated quickly in order to allow a quick response to a changing situation, the strategy makes use of offline and online computation in order to maximize the analytical efficiency. The aim is to carry out as much analysis as possible before the mission begins (offline) and to hence reduce the amount of computation required once the mission is underway (online), therefore minimizing the time taken to make a decision[3].

Binary Decision Diagram (BDD) models have been found to offer potential for performing the real-time mission unreliability analysis that is needed in the decision making process, due to their ability to provide accurate, fast updated unreliability analysis.

Research into the use of BDDs to analyze PMS can be categorized as following two approaches. In the first approach, presented in[4 5], the BDDs representing individual phase failures are rapidly connected together to represent the mission failure logic without accounting for dependencies, which are dealt with during the quantification process. The benefit of this approach is that it allows quantification to begin almost immediately. However, the quantification process itself is inefficient due to the fact that dependencies must be accounted for during implementation. In the second approach, the BDDs are built in such a way that the dependencies are considered during construction, meaning that the quantification process is more efficient than in the first approach. However, the start of the quantification process is delayed when compared to the first approach due to the more complex BDD construction process. In[5], researchers developed BDD construction rules using a phase algebra and made changes to the quantification process to analyze the reliability of PMS using BDDs for the first time. Work presented in[6 7] improves the analysis efficiency of the BDD model in[5] by replacing modules, subtrees whose basic events do not occur anywhere else in the phased mission fault tree, with module events in order to simplify the fault tree structure.[7 8] extended the BDD model presented in[5] to allow mission reliability to be analyzed for PMS containing components that can fail in multiple modes.

Despite the research already conducted, further work is required in order to achieve the speed of quantification needed to make real-time decisions when considering PMS with multiple failure mode components. Tests on a set of benchmarks have indicated that the second approach is the most efficient[9]. Therefore, the investigation presented here will focus on this second approach.

BDD construction initially requires variable ordering and this can have a big impact on BDD size[10 11] and hence the time taken to perform quantitative analysis. Thus if the reliability analysis of a PMS is to be used to support a decision making process, the variable ordering scheme used to construct the BDD can directly affect how quickly decisions can be made as to the best next course of action.

In this paper, an ordering strategy is developed specifically for application within a decision making process. Its aim is to allow the fastest possible calculation of updated unreliability no matter what the future mission configuration. The work presented in this paper also aims to address the requirement for fast, accurate calculation of updated unreliability, by proposing two amendments to the BDD model presented in[7] in order to correct inaccuracies that have been highlighted in past research[8 12], and proposing a more efficient quantification method for the BDD model presented in[8].

The remainder of this paper is organized as follows. Section 2 reviews the fundamentals of BDD and PMS analysis. Section 3 proposes a new ordering scheme, which has features that make it ideal for use in a decision making process. Section 4 reviews existing BDD models and proposes changes, which are intended to correct inaccuracies or improve the efficiency of quantification. Section 5 demonstrates the impact of the variable ordering schemes and BDD model developments using a set of randomly-generated PMS. Finally, conclusions are drawn to highlight the potential impact of this research on the use of reliability analysis to support decision making in a PMS.

2 Background

2.1 BDD

A BDD is a directed acyclic graph based on Shannon's Decomposition[13]. Fault trees can be converted into BDD format[13] to allow efficient analysis of system failures. A BDD node is represented by an if-then-else (*ite*) structure:

$$F = ite < x, F1, F0 >= x \cdot F1 + \bar{x} \cdot F0. \tag{1}$$

If x occurs, then the BDD is traversed along the 1-branch to node $F1$, otherwise, it is traversed along its 0-branch to node $F0$. The BDD is traversed in this way until a terminal node, with a value of 0 or 1, is reached.

In order to construct a BDD, a variable ordering scheme is first defined. The variable ordering can significantly impact the BDD size[10], which is measured by the number of distinct non-terminal nodes the BDD contains, and hence the time taken to perform analysis.

Two BDD nodes, $F = ite < x, F1, F0 >$ and $G = ite < y, G1, G0 >$, where x appears before y in the ordering scheme or the two variables are identical ($x \leq y$), are combined as follows:

$$F \Diamond G = \begin{cases} ite < x, F1 \Diamond G, F0 \Diamond G > & \text{if } x < y \\ ite < x, F1 \Diamond G1, F0 \Diamond G0 > & \text{if } x = y \end{cases} \tag{2}$$

where $\Diamond$ is the Boolean operator, $AND(+)$ or $OR(\cdot)$.

Figure 1 shows the impact of two variable ordering schemes on BDD size for an example fault tree, with the BDDs constructed using Equation 2.

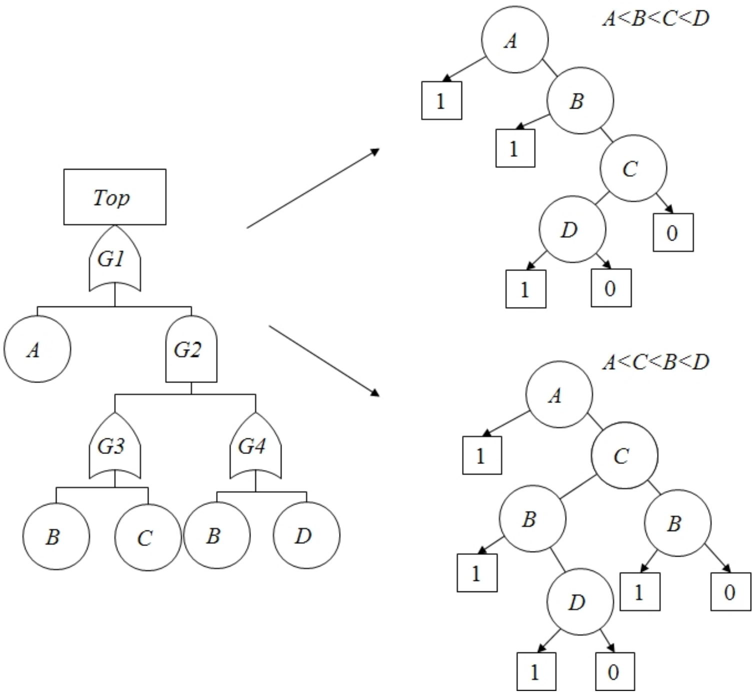

Figure 1: An illustration of the effect of variable ordering on BDD size.

2.2 Modelling PMS failures

A PMS performs a number of tasks in sequence in order to carry out a mission. To analyze the reliability of the mission, fault trees are used to model the failure of the system to perform each task. These fault trees can then be combined to model phases and overall mission failure.

The logic expression for the failure of phase i, F_i, is represented by a fault tree whose basic events are associated with a phase index subscript notation to represent the occurrence of the component failure within a certain time duration and superscript notation to represent the failure mode in which the component fails[14]. Using this notation, the failure of component A in mode q between the start of the mission and the end of phase i is denoted as A_{0i}^q. The fault tree representing mission failure, F_{miss}, is represented by an OR gate with inputs F_i, where $i = 1, 2, \cdots, n$ and n is the total number of phases in the mission[9].

The mission unreliability, Q_{miss}, is the top event probability of the fault tree representing F_{miss}:

$$Q_{miss} = P(F_{miss}) = P(F_1 + F_2 + \cdots + F_n). \tag{3}$$

The probability of the conditional failure of phase i (failure in phase i conditional on the success of the previous $i-1$ phases), $P(ph)_i$, is calculated using the probability of $F_1 + F_2 +$

$\cdots + F_i$[6].

$$\begin{aligned} P(ph_1) &= P(F_1), \\ P(ph_i) &= P(F_1 + F_2 + \cdots + F_i) - P(F_1 + F_2 + \cdots + F_{i-1}). \end{aligned} \tag{4}$$

In order to perform the analysis efficiently, BDDs can be constructed to represent the logic expression for phase and mission failures with dependencies between variables dealt with as necessary during construction and quantification[5 7 8].

3 A Novel Variable Ordering Scheme

3.1 The Motivation for the Best Order Interleaving Scheme (BOI)

When analyzing a PMS, a variable will contain information about which component it relates to and in which failure mode and phase it fails. Therefore, variables must be ordered at three levels[15 11]: component, phase and failure mode level. There are two types of phase ordering: forward ordering considers variables in order of their phase index and backward ordering considers variables according to the reverse of their phase index[16]. The only requirement for failure mode ordering is that the ordering of failure modes in all phases is consistent. Component level ordering is the most complex aspect of variable ordering since the number of components, if large, can lead to many alternative variable lists. Several different component-level ordering schemes have been investigated and can be applied to mission failure fault trees where phase and failure mode indices are neglected (referred to as 'don't care' fault trees), as shown in Appendix A.

However, when events (such as component failures or changing environmental conditions) render the reliability of the current mission unacceptable and make it obligatory for mission reconfiguration to take place, it is necessary to assess and compare the reliability of alternative missions that involve different functionality from the original configuration. This changing system functionality will lead to a requirement for different task fault trees to be considered during the analysis. These fault trees will have different structures to those originally considered and may also contain different basic events. Therefore, if using existing ordering schemes, variables must be reordered given the alternative mission configuration, BDDs constructed for the relevant task failure fault trees and the completed mission phases and analysis of the proposed mission phases conducted. This is potentially inefficient, since the variable ordering and BDD construction must be carried out online, while the mission is in progress, as shown in Figure 2, before quantification can be carried out.

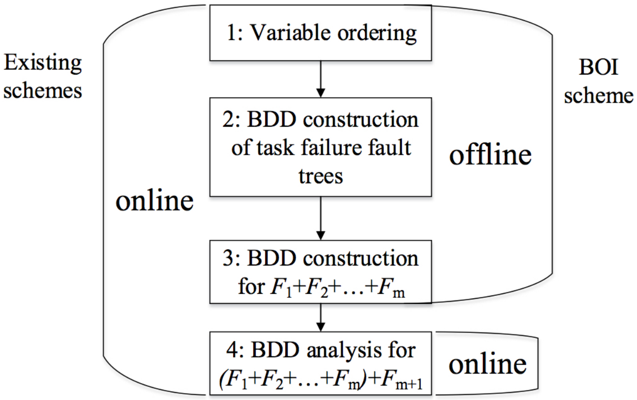
Existing schemes
online
1: Variable ordering
2: BDD construction of task failure fault trees
3: BDD construction for $F_1+F_2+\ldots+F_m$
4: BDD analysis for $(F_1+F_2+\ldots+F_m)+F_{m+1}$
BOI scheme
offline
online

Figure 2: Illustration of the source of the efficiency advantage of BOI over existing ordering schemes, considering the steps required to compute the unreliability of new mission configurations after m phases of the original mission.

Ideally, if taking advantage of the offline-online strategy introduced in[3], online computation and analysis time would be minimized by carrying out as much analysis as possible offline. The Best Order Interleaving Scheme (BOI) is introduced in order to allow variable ordering and BDD construction to be carried out offline, before the mission begins, as shown in Figure 2. The goal is to promote efficiency and to ensure that a decision can be made about the best next course of action without unnecessary delay.

BOI arranges variables according to the structures of the fault trees representing the failures of all tasks that a PMS is capable of rather than the structure of the fault tree representing mission failure. Variables are ordered before the mission starts and the variable order list remains unchanged no matter how the mission configuration changes. The principles of the BOI scheme allow the online-offline strategy to be adopted and hence maximize the amount of computation that is carried out offline. This will save a large amount of effort to re-order variables and construct BDDs when performing reliability analysis for alternative mission configurations when the mission is underway.

3.2 Application of BOI

BOI uses an interleaving technique, which was first introduced in[17] to provide an optimal order list for multiple output combinatorial circuits. The technique combines the variable order lists that lead to the smallest BDD size for each task failure fault tree to obtain an overall list for all possible variables.

The ordering schemes described in Appendix A can be classified into two groups according to the principles followed when applying them: group 1: Schemes 1, 5, 6, and 8 and group 2: Schemes 2, 3, 4, and 7.

In the group 1 schemes, variables are ordered in a global range according to an assigned value of certain parameters. In the group 2 schemes, variables are explored in a depth-first manner, i.e., variables below a gate are fully allocated before the exploration of another gate. This common principle of group 2 schemes provides a basis for applying the interleaving technique. Therefore, the BOI scheme uses group 2 schemes to obtain optimal variable order lists for individual task failure fault trees so that they can be integrated using the interleaving technique.

The principles of the BOI scheme are:

1. Identify task failure fault trees for the PMS.

2. Use the four schemes to analyze each of the task fault trees, thus obtaining 4 BDDs for each fault tree.

3. Compare the sizes of the 4 BDDs and select the optimal ordering scheme for each, i.e., that which leads to the smallest BDD size. Cache the optimal order list and the smallest BDD size for each fault tree.

4. Prioritize the order lists according to decreasing size of the smallest BDD for each fault tree.

5. Interleave variables in the order lists using the interleaving technique described below until all variables are included in the final list[17]:

   (a) For the first variable in the order list, if the variable is already in the final list, then do nothing; otherwise, insert it at the beginning of the final list.

   (b) For the other variables in the order list, check whether the variable, (assume without loss of generality) A, is in the final list, if so, do nothing; otherwise, identify the position in the final list of the variable B that is immediately in front of variable A in the current order list and insert variable A immediately after variable B in the final list.

Consider for instance a system that can perform 4 tasks. The fault trees representing the failure of the system to complete these tasks are shown in Figure 3. In a mission, M_A, the system is required to perform task 1, task 2 and task 3 in sequence. The failure of mission

M_A is represented by the fault tree in Figure 4, where the variables associated with the fault tree basic events encode information relating to the phase (superscript) and failure mode (subscript) in which a component fails. For example, A^q_{0i} represents component A failing in failure mode q between the start of the mission and the end of mission phase i.

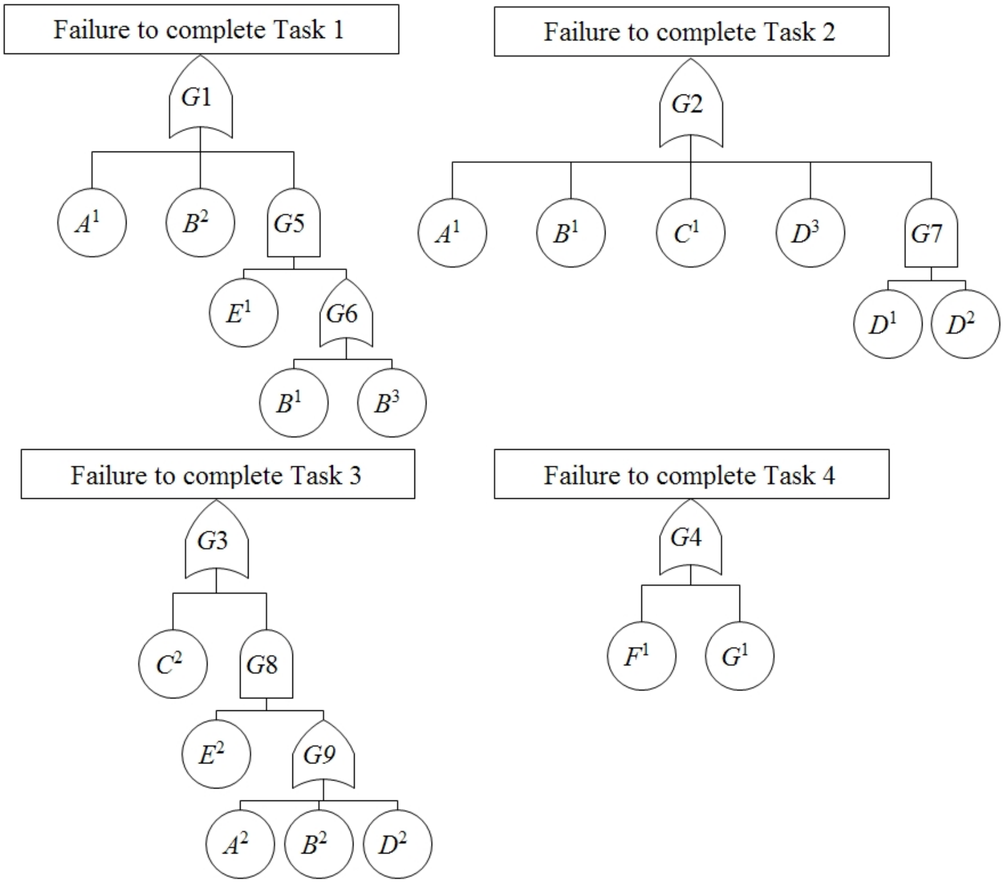
Failure to complete Task 1
G1
A1
B2
G5
E1
G6
B1
B3
Failure to complete Task 2
G2
A1
B1
C1
D3
G7
D1
D2
Failure to complete Task 3
G3
C2
G8
E2
G9
A2
B2
D2
Failure to complete Task 4
G4
F1
G1

Figure 3: Fault trees representing the failure of a system to complete 4 tasks.

Although task 4 is not included in the mission, variables are ordered according to the fault tree structures of all tasks the system can perform, as shown in Figure 3, so that when considering an alternative mission that includes task 4, there is no need to re-order variables or reconstruct BDDs for previous successfully completed phases.

Applying the four schemes to each of the task failure fault trees leads to the following order lists and associated BDD size for each fault tree: Task 1: $B < A < E$ with size 8, Task 2: $D < A < B < C$ with size 4, Task 3: $C < E < A < B < D$ with size 6, Task 4: $F < G$ with size 2. Thus, the order lists are considered in the sequence Task 1, Task 3, Task 2 and Task 4. The final variable order list is created as follows. First, the order list of Task 1 is copied to the final list to give $B < A < E$. Variables from Task 3 that are not

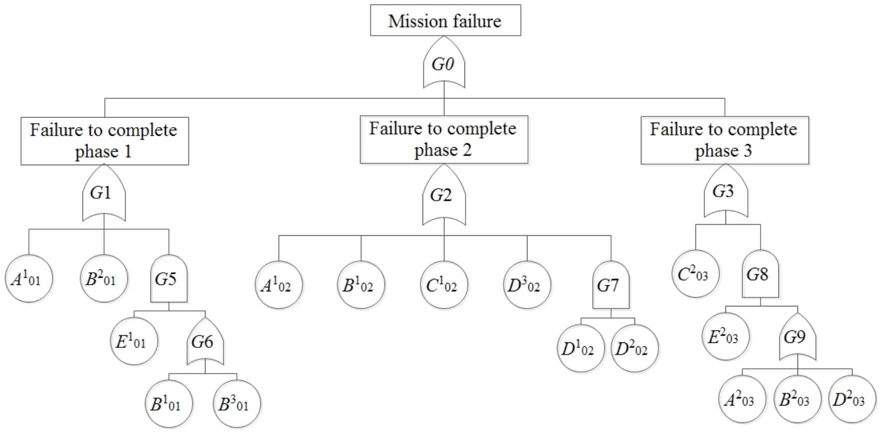

Figure 4: The fault tree representing the failure of the system to complete mission M_A.

yet in the final list must then be added. C is the first variable in the order list for Task 3. Therefore, it is allocated at the start of the final list. E, A, and B are already in the final list, meaning no action is required. The final variable, D, appears after B in Task 3 so it is inserted immediately after B in the final list to obtain $C < B < D < A < E$. The final variable order list is complete after the variables from Task 2 and Task 4 have been added: $F < G < C < B < D < A < E$.

4 Improving the Efficiency of PMS Reliability Analysis Using BDDs

Two BDD models have been developed to address the dependencies that appear in PMS with multiple failure mode components during BDD construction. The DEP-BDD model presented in[7] takes account of the dependencies that arise due to the mission phases and multiple failure modes using dependence and phase algebra. The model has been shown to give inaccurate results[8 12] but instead of correcting the DEP-BDD model, the researchers who discovered the inaccuracy developed a new model, which uses a forward phase ordering for BDD construction and an Implicant Tree method for quantification. This model will be referred to here as the Forward-BDD model.

The remainder of this section details suggested improvements to existing PMS BDD models:

- Two amendments are proposed to the DEP-BDD analysis to correct the previously-observed inaccuracies;

- A more efficient quantification method is proposed to replace the quantification method of the Forward-BDD model.

4.1 Amending and Improving the DEP-BDD Model

4.1.1 Analytical Inaccuracies

The failure probability of the mission modelled by the fault tree given in Figure 5 can be shown to be[18]:

$$P(Top) = P(A^2_{01}) + P(A^1_{02}). \tag{5}$$

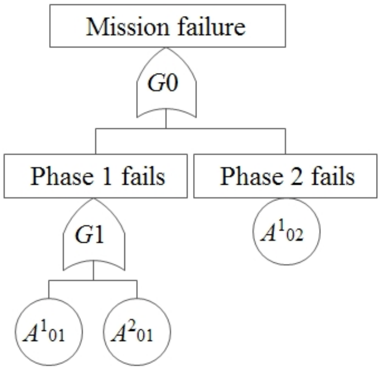

Figure 5: Example mission fault tree.

Using the DEP-BDD model (as presented in Appendix B), variables in the example fault tree shown in Figure 5 are ordered as: $A^1_{02} < A^2_{01} < A^1_{01}$ and the constructed BDD is shown in Figure 6.

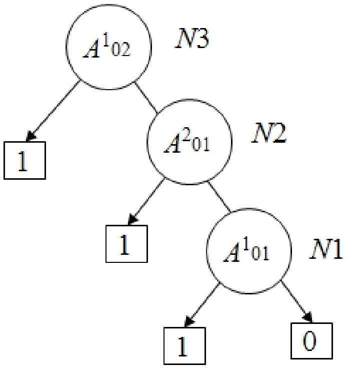

Figure 6: The BDD representing the mission failure conditions show in Figure 9 obtained using the DEP-BDD model.

The root node probability evaluated using the DEP-BDD evaluation, Equation 15, is:

$$P(Top) = P(A_{01}^{2}) + P(A_{01}^{1}) + P(A_{02}^{1}), \tag{6}$$

which is incorrect. The inaccuracy arises due to the node $N1$, which is redundant[12].

In Section 4.1.2 and Section 4.1.3, two modifications are proposed in order to correct the DEP-BDD analysis by eliminating redundant nodes from the BDD and hence allowing accurate quantification to be performed. These two modified models are referred to here as Model 1 and Model 2 and are quantified using the same method that is used in the original DEP-BDD model.

4.1.2 Applying a Reduction Process (Model 1)

Model 1 retains the variable ordering of the DEP-BDD model and corrects the quantification process by adding a reduction process to simplify the construction of two nodes when variables relate to the same component but different failure modes to ensure BDDs remain in a compact format.

- The calculation of L_1 in the DEP-BDD model is modified so that L_1 is the first node with a variable relating to a component other than x or relating to the same component and failure mode as x encountered on the traversal down the 0-branches of the node starting from G. This is because $x = 1$ implies variables related to another failure mode of the same component must be equal to 0 whereas this is not true for other cases.

- The calculation of L_0 can be simplified to $G0$, since using the construction rules of Model 1, the 0-branch of node G always links two variables that relate to different failure modes of the same component or that relate to different components, meaning that when $x = 0$, $L_0 = (y \cdot G1 + \bar{y} \cdot G0)_{x=0} = (0 \cdot G1 + 1 \cdot G0)_{x=0} = (G0)_{x=0} = G0$ always applies.

- A reduction is carried out when computing the combination of two nodes whose variables relate to an identical component and failure mode. The process involves traversing down the 0-branch of the newly-created node and replacing any node with a variable relating to an identical component and failure mode as the newly-created node by its 0-branch.

This reduction process eliminates redundant nodes in the BDD, keeping the BDD in a simplified form. Since there are no redundant nodes, the DEP-BDD quantification process described in[7] will yield accurate results. The speed of the analysis should also reduce due to the reduction in the BDD size.

4.1.3 Amending Variable Ordering (Model 2)

Model 2 corrects the DEP-BDD analysis by changing the variable ordering and adopting the same L_1 and L_0 calculation as used in Model 1. Model 2 requires the variables to be ordered firstly according to failure mode level and then phase level. This is the only change from the Model 1 analysis. However, by considering failure mode level first, the phase dependency operation automatically eliminates redundant nodes from the BDD. No additional reduction process is needed.

In common with Model 1, the elimination of redundant nodes in the BDD when it is constructed using Model 2 means that applying the DEP-BDD quantification process will yield accurate results.

4.1.4 Example

For the fault tree shown in Figure 5, the BDD constructed using Model 1 before reduction is shown in Figure 6. Traversing the BDD from $N3$ along its 0-branch, it can be seen that the variable of node $N1$, A^1_{01}, relates to the same failure mode of the same component as the variable of the root node $N3$, A^1_{02}. This means that $N1$ is a redundant node and thus is replaced by its 0-branch, 0, leading to the BDD shown in Figure 7.

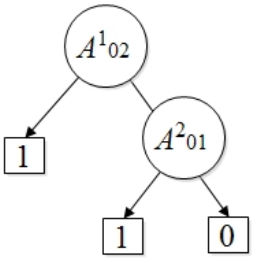

Figure 7: BDD obtained using Model 1.

Using the DEP-BDD quantification process gives:

$$P(Top) = P(A^2_{01}) + P(A^1_{02}). \tag{7}$$

Using Model 2, the variables are ordered as: $A_{01}^2 < A_{02}^1 < A_{01}^1$, with A_{01}^2 being listed earlier than A_{02}^1 because Model 2 considers dependency between failure modes before dependency between phases. Using the same construction rules as those given for Model 1, the BDD shown in Figure 8 is obtained and the DEP-BDD quantification process gives:

$$P(Top) = P(A_{01}^2) + P(A_{02}^1). \tag{8}$$

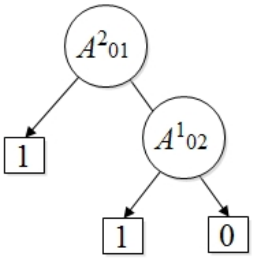

Figure 8: BDD obtained using Model 2 and Forward-BDD model.

It can be seen by comparing Equation 5 to Equation 8 that Models 1 and 2 correct the observed inaccuracy of the DEP-BDD model and yield accurate results following quantification.

4.2 Amending the Forward-BDD method (Model 3)

The Forward-BDD model[8] was developed as an accurate alternative to the DEP-BDD for analyzing PMS with multiple failure mode components. Variables are ordered according to failure mode level before phase level and it uses forward phase level ordering.

4.2.1 The Forward-BDD Model

The Forward-BDD model uses the following rules to compute the operation between two nodes, $F = ite < x, F1, F0 >$ and $G = ite < y, G1, G0 >$. Suppose that $x \leq y$: $F \diamond G =$

$$\begin{cases} ite < x, F1 \diamond G1, F0 \diamond G0 > & x = y \\ ite < x, F1 \diamond G, F0 \diamond G > & cp(x) \neq cp(y) \\ ite < x, F1 \diamond L_1, F0 \diamond G > & cp(x) = cp(y), fm(x) \neq fm(y) \\ ite < x, F1 \diamond G1, F0 \diamond G > & cp(x) = cp(y), fm(x) = fm(y) \end{cases} \tag{9}$$

where $L_1 = (G0)_{x=1}$ is the first node with variable relating to a component other than x encountered during a traversal down the 0-branch of the BDD starting from G.

For a newly-created node as shown in Figure 9, a reduction rule is introduced to remove redundant nodes[8]. If $cp(x) = cp(z)$, $fm(x) = fm(z)$, and $G1 = I1$, then node G is replaced by its success branch $G0$.

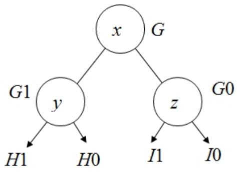

Figure 9: A general node in a PMS BDD.

For the fault tree in Figure 5, variables are first ordered by backward failure mode and then forward phase index: $A^2_{01} < A^1_{01} < A^1_{02}$. By applying the Forward-BDD construction rules and reduction process, the BDD obtained is the same as that obtained using Model 2, as shown in Figure 8.

The Implicant Tree method developed in[8] allows quantification of the BDDs constructed using the Forward-BDD model by constructing a dependency free data structure, the Implicant Tree.

However, the Implicant Tree is a new data structure that must be constructed from the BDD prior to quantification. This is likely to add significant computation time to the quantification procedure, particularly when the BDD is large. This is not a desirable feature for a phased mission analysis technique that is to be used as part of a decision making process. Therefore, a new method is developed here to quantify the Forward-BDDs directly. The efficiency of the proposed quantification method is demonstrated in Section 5.1.

4.2.2 Proposed Quantification Method for Forward-BDDs (Model 3)

In Model 3, the 1-branch always links two variables that relate to different components and the 0-branch can either link two variables that relate to different components or two variables that relate to the same component due to the Forward-BDD construction rules[8].

The proposed quantification method is based on the phase algebra in Table 1, which is used to deal with dependencies across phases[5] and the dependency algebra in Table 2, which is introduced to deal with dependencies between multiple failure modes[7], of two variables relating to the same component.

Table 1: Phase algebra for two variables (A^p_{0i} and A^p_{0j}) relating to the same component and failure mode but with different phase index ($i < j$)

$A^p_{0i} \cdot A^p_{0j} = A^p_{0i}$	$\overline{A^p_{0i}} \cdot \overline{A^p_{0j}} = \overline{A^p_{0j}}$
$\overline{A^p_{0i}} + \overline{A^p_{0j}} = \overline{A^p_{0i}}$	$A^p_{0i} + A^p_{0j} = A^p_{0j}$
$\overline{A^p_{0i}} + A^p_{0i} = 1$	$A^p_{0i} \cdot \overline{A^p_{0j}} = 0$

Table 2: Dependency algebra for two variables (A^p_{0i} and A^q_{0j}) relating to the same component but different failure modes ($p \neq q$).

$A^p_{0i} \cdot A^q_{0j} = 0$	$\overline{A^p_{0i}} \cdot A^q_{0j} = A^q_{0j}$
$A^p_{0i} \cdot \overline{A^q_{0j}} = A^p_{0i}$	$A^p_{0i} + \overline{A^q_{0j}} = \overline{A^q_{0j}}$
$\overline{A^p_{0i}} + A^q_{0j} = \overline{A^p_{0i}}$	$A^p_{0i} + A^q_{0j} = A^p_{0i} + A^q_{0j}$

For a node G shown in Figure 9, the quantification method is given by: $P(G) =$

$$\begin{cases} p(x) * P(G1) + [1 - p(x)] * P(G0) & \text{case 1} \\ P(G0) + p(x) * [P(G1) - P(I1)] & \text{case 2} \\ P(G0) + p(x) * [P(G1) - P((I0)_{x=1})] & \text{case 3} \end{cases} \tag{10}$$

where $*$ is the probability multiplication, $(I0)_{x=1}$, is the first node with variable relating to a component other than x encountered during the traversal down the 0-branch of node $G0$ and the conditions relating to each case are:

case 1: $cp(x) \neq cp(z)$

case 2: $cp(x) = cp(z)$ and $fm(x) = fm(z)$

case 3: $cp(x) = cp(z)$ and $fm(x) \neq fm(z)$

The proof of Equation 10 is given below. Since $cp(x) \neq cp(y)$ in all cases, $P(x \cdot G1) = p(x) * P(G1)$. Three cases must then be considered.

1. When $cp(x) \neq cp(z)$:

$$\begin{aligned} P(G) &= P(x \cdot G1 + \bar{x} \cdot G0) \\ &= P(x \cdot G1) + P(\bar{x} \cdot G0) \\ &= p(x) * P(G1) + p(\bar{x}) * P(G0) \\ &= p(x) * P(G1) + (1 - p(x)) * P(G0) \end{aligned}$$

2. When $cp(x) = cp(z)$ and $fm(x) = fm(z)$,

$$\begin{aligned} P(\bar{x} \cdot G0) &= P(\bar{x} \cdot (z \cdot I1 + \bar{z} \cdot I0)) \\ &= P(\bar{x} \cdot z \cdot I1 + \bar{x} \cdot \bar{z} \cdot I0) \\ &\text{(according to the phase algebra, } \bar{x} \cdot z = z - x \cdot z \\ &\text{and } \bar{x} \cdot \bar{z} = \bar{z}) \\ &= P((z - x \cdot z) \cdot I1 + \bar{z} \cdot I0) \\ &= P(z \cdot I1 + \bar{z} \cdot I0 - x \cdot z \cdot I1) \\ &\text{(according to the phase algebra, } x \cdot z = x \text{)} \\ &= P(G0 - x \cdot I1) \\ &= P(G0) - p(x) * P(I1) \end{aligned}$$

Here, $(I1)_{x=1} = I1$ always holds because the variable of $I1$ is always different to x^8. Substituting $P(\bar{x} \cdot G0)$ into $P(G)$ gives:

$$\begin{aligned} P(G) &= P(x \cdot G1 + \bar{x} \cdot G0) \\ &= P(x \cdot G1) + P(\bar{x} \cdot G0) \\ &= p(x) * P(G1) + P(G0) - p(x) * P(I1) \\ &= P(G0) + p(x) * [P(G1) - P(I1)] \end{aligned}$$

3. When $cp(x) = cp(z)$ and $fm(x) \neq fm(y)$,

$$\begin{aligned} P(\bar{x} \cdot G0) &= P(\bar{x} \cdot (z \cdot I1 + \bar{z} \cdot I0)) \\ &= P(\bar{x} \cdot z \cdot I1 + \bar{x} \cdot \bar{z} \cdot I0) \\ &\text{(according to the dependence algebra, } \bar{x} \cdot z = z \text{ and} \\ &\bar{x} \cdot \bar{z} = \bar{z} - x \cdot \bar{z}) \\ &= P(z \cdot I1 + \bar{z} \cdot I0 - x \cdot \bar{z} \cdot I0) \end{aligned}$$

$$\begin{aligned}&(\text{according to the dependence algebra, } x \cdot \bar{z} = x)\\ &= P(G0 - x \cdot I0)\\ &= P(G0) - p(x) * p((I0)_{x=1})\end{aligned}$$

Substituting $P(\bar{x} \cdot G0)$ into $P(G)$ gives:

$$\begin{aligned}P(G) &= P(x \cdot G1 + \bar{x} \cdot G0)\\ &= P(x \cdot G1) + P(\bar{x} \cdot G0)\\ &= p(x) * P(G1) + P(G0) - p(x) * P((I0)_{x=1})\\ &= P(G0) + p(x) * (P(G1) - P((I0)_{x=1}))\end{aligned}$$

4.2.3 Example

Quantification of the BDD shown in Figure 8 using the proposed quantification needs only to consider the case when $cp(x) = cp(z)$ and $fm(x) \neq fm(z)$ and gives the same result as Equation 5:

$$P(top) = P(A^2_{01}) + P(A^1_{02}). \tag{11}$$

5 Performance Results and Comparison

The accuracy of the DEP-BDD model and the Forward-BDD model were compared in[8], where it was shown that the quantification results obtained using the Forward-BDD model were accurate, unlike those obtained using the DEP-BDD model. The mission failure probabilities obtained using the three new models presented here are identical to those obtained using the Forward-BDD model, thus the four models are compared solely in terms of analytical efficiency.

Two efficiency measures are used to assess the performance of the BDD models: the size of the BDD representing mission failure (number of BDD nodes), and the mission analysis time. This mission analysis time is considered in the context of mission reconfiguration, when updated failure probabilities must be calculated for a number of alternative mission configurations and a decision made as to the best next course of action. Since this is the case, the time taken to order variables and to construct and quantify mission failure BDDs must be included. In order to allow decisions to be made quickly, the shorter the mission analysis time, the better.

Software was written to generate benchmark fault trees of varying size and structure using the method presented in[19]. The structural features of a fault tree are mainly decided by the following parameters: the minimum and maximum number of gates, the percentage of gate inputs in each fault tree layer and the maximum number of component failure modes. A number of fault tree sets were produced and combined at random in order to create 100 mission profiles, which could then be used to test the efficiency of the four BDD models.

5.1 Comparison of Model Efficiency

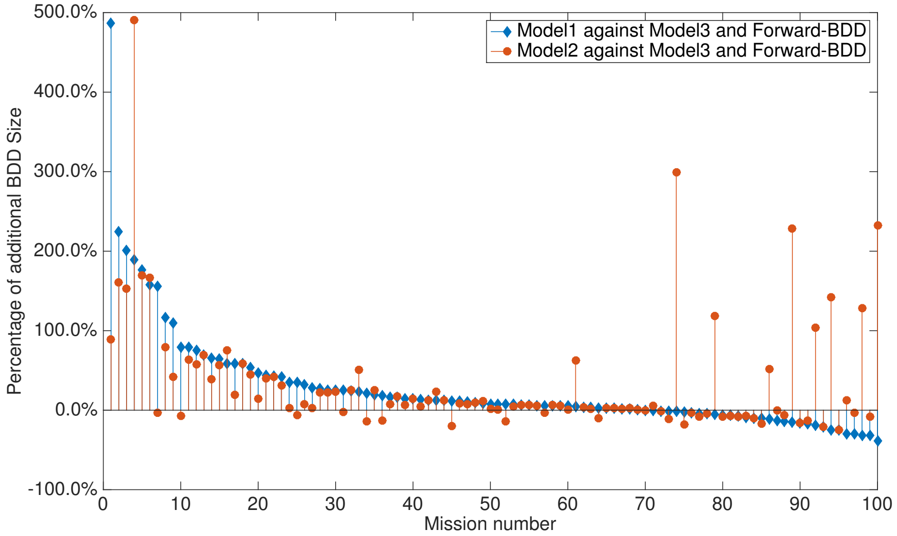

Figure 10: The percentage of additional BDD sizes generated using Model 1 and Model 2 compared with the BDD sizes generated using Model 3 and Forward-BDD model

Table 3 shows the number of missions for which each model produces a BDD smaller than those produced by the other models. Model 3 shares its BDD construction rules with the Forward-BDD model and therefore shares its BDD sizes. These models are seen to generate smaller BDDs than the other two models for 59 PMS. For the other 41 cases, the smallest BDD is produced by either Model 1 or Model 2. Figure 10 shows a comparison of the size of the BDDs generated by Model 1 and Model 2 relative to those generated by Model 3 and the Forward-BDD Model for each of the PMS considered.

Table 3: Performance comparison of BDD models in terms of BDD sizes

BDD model	Model 1	Model 2	Model 3 and Forward-BDD
No. of missions	23	18	59

Table 4 shows the number of missions for which the analysis of each model takes less time

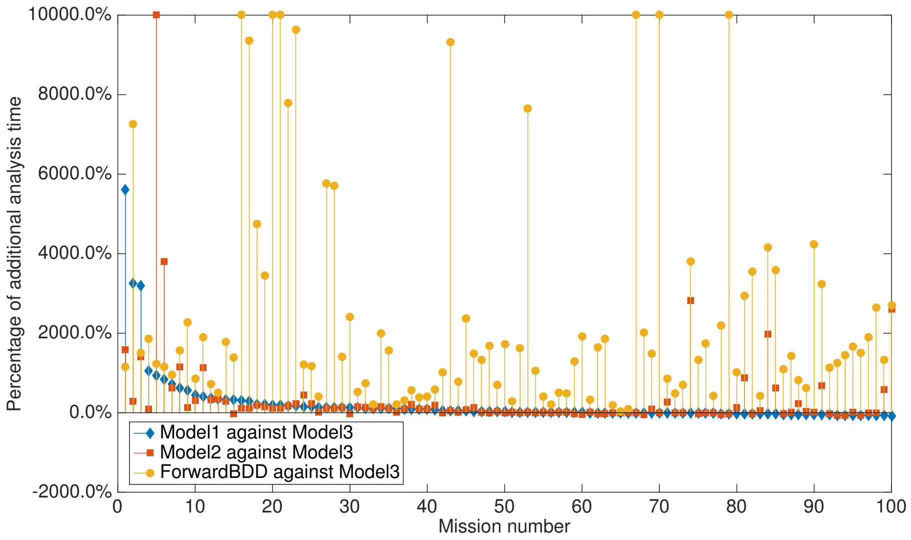

Figure 11: The percentage of additional analysis time generated using Model 1, Model 2 and Forward-BDD model compared with the analysis time generated using Model 3

than the analysis of the same mission using other models. In 51 cases, the fastest analysis comes when using Model 3. For the other 49 cases, Model 1 or Model 2 result in the fastest analysis (the Forward-BDD Model never results in the fastest analysis). Figure 11 shows a comparison of the time taken to perform the analysis when using Model 1, Model 2 and the Forward-BDD Model relative to the time taken when using Model 3 for each of the PMS considered.

Therefore, improved efficiency can be expected when using Model 1, 2 and 3. Of the three developed BDD models, Model 3 can be seen to generally perform better than Model 1 and Model 2, since it leads to the smallest analysis time for around 50% of the missions.

Table 4: Performance comparison of BDD models in terms of analysis time

BDD model	Model 1	Model 2	Model 3	Forward-BDD
No. of missions	30	19	51	0

Table 5 gives the efficiency improvement of the three developed BDD models over the Forward-BDD model computed by the average reduction in mission analysis time. The time taken to perform analysis when using the Forward-BDD model is longer in all cases than the time taken to perform analysis when using the other models. All of the new BDD models show greatly reduced analysis time in comparison to the Forward-BDD model. A particularly noteworthy result is seen for Model 3, which, despite using the same BDD structure as the

Forward BDD model, results in an analysis time that is on average 90.09% less than that required for the Forward BDD model.

Table 5: Analysis of the efficiency improvement (percentage of time reduced) of the three developed BDD models over the Forward-BDD model

BDD models	Model 1	Model 2	Model 3
Average improvement	75.28%	54.12%	90.09%

5.2 Comparison of Variable Ordering Schemes

Since Model 3 proved to be the most efficient BDD model for the tested phased missions, it is used to analyze the efficiency of the variable ordering schemes. Using Model 3, each ordering scheme is used during the construction of mission failure BDDs for the 100 test mission profiles. Table 6 shows the number of missions for which each scheme produced the smallest BDD and also the number of missions for which each scheme resulted in the lowest mission analysis time. The table shows that it is possible for any scheme to produce the smallest BDD. When considering analysis time, Scheme 6 and Scheme 8 never result in the lowest time, since the variable ordering time required when applying these schemes takes longer than that required when applying the other ordering schemes. Scheme 5 results in the smallest mission failure BDD in 30 out of the 100 cases and the lowest analysis time in 48 out of the 100 cases. Therefore, Scheme 5 would appear to be most likely to offer performance advantages in terms of reduced analysis time when compared to the other ordering schemes. It should give the greatest chance of producing the lowest analysis times when calculating updated probabilities.

Table 6: Performance comparison of the nine ordering schemes measured by size of BDD and analysis time for mission unreliability

	Scheme 1	Scheme 2	Scheme 3	Scheme 4	Scheme 5	Scheme 6	Scheme 7	Scheme 8	BOI
BDD size	4	4	10	4	30	10	19	16	3
Analysis time	8	11	22	6	48	0	4	0	1

5.3 The Performance of BOI

The proposed BOI scheme can be seen to perform inefficiently in comparison to the other ordering schemes when analyzing mission unreliability. However, this scheme was specially developed with a view to enabling the efficient use of the reliability analysis of PMS in a decision making tool. It is expected to improve analysis efficiency when calculating the unreliability of possible alternative missions that must be considered when the unreliability of the current mission drops to an unacceptable level and alternatives must be considered. In particular, this improvement in analysis time is likely to be observed when calculating the unreliability of the first phase that involves performing a task that was not a part of the original mission, since the fault tree representing the failure to complete this task is most likely to contain basic events that do not appear in the current mission fault tree. This means that variable re-ordering is required before any further analysis can be performed and that the analysis of the phases completed up to the current point in the mission must be repeated for this new ordering (as illustrated in Figure 2).

Table 7 shows the average efficiency advantages (percentage reduction in analysis time) derived when using BOI in preference to the other ordering schemes to analyze the unreliability of the first altered phase, in an alternative mission configuration for two different situations: when the first altered phase is the third phase in the alternative mission and when it is the penultimate phase. The average efficiency advantage of the BOI scheme over the other schemes varies from 39.33% to 95.13%. In all cases, BOI leads to a lower unreliability analysis time for the first altered phase in the alternative mission, thus demonstrating the potential for its use in supporting fast decision making in PMS.

Table 7: Average relative advantage of BOI compared to the eight ordering scheme when analyzing the first altered phase in the alternative mission

	Scheme 1	Scheme 2	Scheme 3	Scheme 4	Scheme 5	Scheme 6	Scheme 7	Scheme 8
The third phase	47.96%	40.19%	39.33%	69.95%	48.40%	83.21%	76.44%	95.13%
The penultimate phase	45.42%	49.13%	44.04%	56.62%	43.47%	63.21%	64.23%	75.62%

6 Case Study

In order to further demonstrate the efficiency advantage of the proposed BDD Model 3 in comparison to the other models, and the advantage of the developed BOI scheme within the context of decision making processes for a real world application, an example search and rescue (SAR) mission to be completed by a UAV is considered and different application scenarios modelled by varying mission configurations are tested.

The UAV is capable of performing six tasks and the failure of each task is represented by a modularised fault tree whose structure is detailed in[6]. The six fault trees have structure information shown in Table 8. It is assumed that the highest acceptable mission unreliability is 0.001. If the unreliability exceeds this value for a mission then alternative missions will considered.

Table 8: Fault tree information for the UAV SAR mission case study[6]

Task fault tree	No. gates (OR, AND)	No. events (Non-repeated)	Max. failure mode
FT1 (Takeoff)	17 (9,8)	19	2
FT2 (Climb)	14 (8,6)	14	2
FT3 (Cruise and Search)	15 (8,8)	22	2
FT4 (Cruise and Detect Submarine)	15 (8,9)	27	2
FT5 (Descend)	14 (8,6)	14	2
FT6 (Land)	17 (9,8)	20	2

The six tasks are arranged in specific sequence and frequency to form a mission configuration. The initial SAR mission failure is shown by the fault tree in Figure 12. Suppose failure probabilities of the basic events follow an exponential distribution[20] with randomly generated failure rates of the order 10^{-6} per hour.

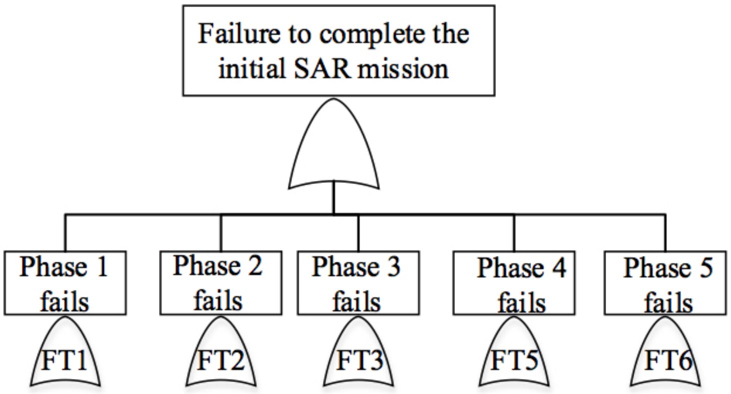

Figure 12: The fault tree representing the failure of the UAV to complete the initial SAR mission

Before the mission starts (off-line), enough time is allowed to analyse the mission unreliability using different ordering schemes and BDD models to select the optimal ones, i.e. those that have the highest potential to provide the most efficient analysis of the updated mission unreliability when the the UAV is in flight. Table 9 shows that for every ordering scheme (comparing the horizontal data), the BDD constructed using Model 3 and the forward-BDD model are smaller than those constructed using Model 1 and Model 2. The analysis performed using Model 3 is always faster than using the other models and thus is selected for future updated mission reliability analysis. The calculated mission unreliability is 0.00016 which is lower than the acceptable threshold of 0.001 and thus the UAV starts the initial mission.

Table 9: Analysis results for the original SAR mission configuration in the off-line stage

Parameter	BDD size				Mission analysis time(s)			
	Model 1	Model 2	Model 3	Forward-BDD	Model 1	Model 2	Model 3	Forward-BDD
Scheme 1	1309	1342	922	922	0.5	0.6	0.5	2.9
Scheme 2	2496	2496	1422	1422	2.5	2.5	0.7	3.4
Scheme 3	2496	2457	1422	1422	2.5	2.5	0.7	3.3
Scheme 4	2820	2714	1791	1791	2.7	2.7	0.8	6.6
Scheme 5	1650	1642	1084	1084	0.8	0.8	0.5	3.2
Scheme 6	6349	6321	1759	1759	11.2	11.2	2.2	19.7
Scheme 7	2766	2741	1788	1788	2.6	2.6	0.9	5.9
Scheme 8	1314	1314	873	873	0.6	0.6	0.5	2.7
BOI	2020	2034	1012	1012	1.8	1.9	0.5	2.8

Suppose during the performance of phase 2 of the SAR mission, the existence of hostile armed submarines is detected which will lead to potential threat that affects a future phase. This means the probabilities of basic events representing the occurrence of an external threat in a later phase fault tree will be updated to a larger value. After a calculation of current mission unreliability based on updated variable probabilities, the mission unreliability is proved to be too high to accept (> 0.001) and thus the original SAR cannot be continued. Therefore, two alternative mission configurations (as shown in Figure 13) are considered to ensure the success of the mission objective.

Both of the missions involve a new task failure fault tree (FT4). Due to the requirement for an instant decision to be made when it is no longer safe to carry on the current mission, it is necessary to calculate the unreliability of the first altered phase and the entire alternative mission[14]. It is seen from Table 10 that the analysis speed when using the BOI scheme is faster than when using the other schemes (comparing vertically).

The results of the case study again support the testing results presented in 5, i.e. Model 3 is more efficient than the other models in terms of analysis speed. For this case study, the BOI scheme is an optimal choice to calculate the unreliability of the first altered phase when an immediate decision is required as to the best next course of action.

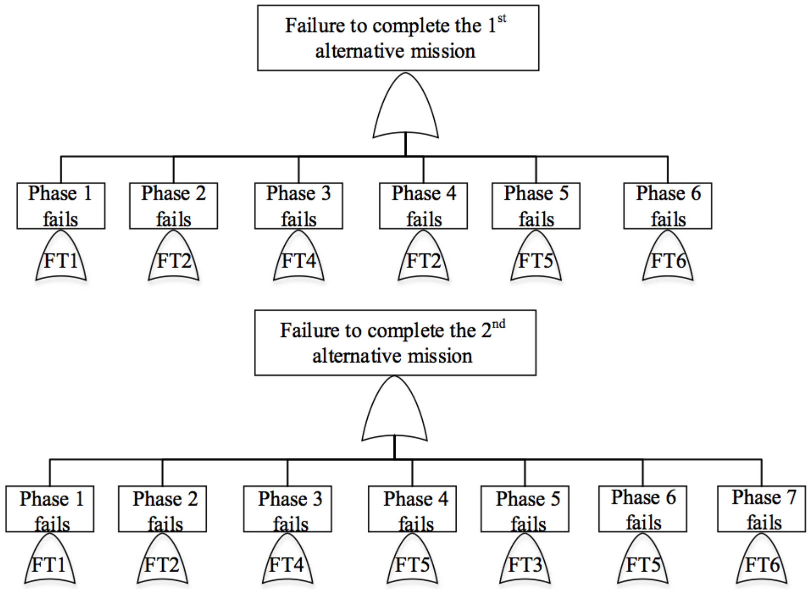

Figure 13: The fault trees representing the failure of the UAV to complete the 1^{st} or the 2^{nd} alternative mission

Table 10: Exact analysis for two mission alternatives

Parameter	BDD size				Analysis time(s)			
Mission	Scenario 1		Scenario 2		Scenario 1		Scenario 2	
Scope	Phase 3	Mission	Phase 3	Mission	Phase 3	Mission	Phase 3	Mission
	Model 1	Model 2	Model 3	Forward-BDD	Model 1	Model 2	Model 3	Forward-BDD
Scheme 1	891	1026	896	1134	0.1	0.6	0.1	0.8
Scheme 2	1375	1519	1375	1851	0.3	1.2	0.3	1.7
Scheme 3	1375	1519	1375	1851	0.3	1.2	0.3	1.7
Scheme 4	1635	1863	1685	2038	0.9	1.6	0.9	1.9
Scheme 5	1048	1233	1124	1404	0.2	0.7	0.3	1.0
Scheme 6	2109	3600	1987	3186	1.6	15.1	1.4	10.3
Scheme 7	2157	2329	1542	2043	1.4	2.8	1.2	2.3
Scheme 8	895	958	932	1081	0.2	0.6	0.2	0.8
BOI	937	1089	960	1190	0.0	0.6	0.0	0.1

7 Conclusion

If a reliability analysis methodology is to be used to support real-time decision making for systems operating phased missions in changing mission environments, it is crucial that the applied methodology can analyze PMS quickly and accurately.

The construction of BDDs initially requires variables to be ordered and how they are ordered can greatly affect the sizes of the constructed BDDs. This paper has tested the efficiency of eight ordering schemes by applying them to 'don't care' fault trees representing PMS failures. A new ordering scheme, BOI, is proposed, which is designed to work efficiently within the decision making process described in the literature. It is specifically developed to enable updated reliability analysis, which is performed when alternative mission configurations must be considered while a mission is in progress, to be performed more quickly.

When there is no time constraint for identifying an appropriate mission alternative (and hence no great urgency to the calculation of mission unreliability), Scheme 5 appears to be the best choice of ordering scheme to use in the analysis, as it has shown to have the highest chance of producing the smallest BDD sizes for mission failure and the lowest mission unreliability analysis time for the tested missions. When time is limited (meaning the configuration of the next phase needs to be decided almost immediately), and particularly in the case when variable reordering would be necessary, the BOI scheme is recommended to be used since it avoids the need for variable reordering and repeated analysis of phases shared by the original and alternative missions, so that the unreliability of the first altered phase of the mission alternative can be quickly computed and an acceptable mission configuration can be decided in the shortest possible time.

In this paper, after reviewing the DEP-BDD analysis, two amendments, Model 1 and Model 2, have been proposed, which correct the previously-observed inaccuracies. The Forward-BDD model is improved by introducing a more efficient quantification method.

As shown by results in Section 5 and the case study, all of the three developed models offer much faster analysis for the PMS with multiple failure modes than the Forward-BDD model. The analysis efficiency advantage of Model 3 over the Forward-BDD model is purely down to the improvement in efficiency that comes from the proposed quantification method, since they use the same BDD construction rules.

Model 3 (or the Forward-BDD model) was shown to have a higher chance of obtaining smaller BDDs compared with the two models that are based on DEP-BDD analysis. Of the three new models, Model 3 was seen to result in the highest percentage reduction in mission analysis time when compared to the other two models for the missions tested. This means that Model 3 would appear to be the most promising to be used when performing reliability analysis for PMS containing components with multiple failure modes to help the system make real-time decisions as to its next course of action in dynamic, rapidly changing mission environments.

Appendix A: Variable Ordering Schemes

Modified Top-down Ordering (Scheme 1)[21]

Variables are ordered as they appear in the ‘don’t care’ fault tree in a top-down, left-right arrangement, with priority given to variables that appear in higher levels of the fault tree. For tied variables, the one that occurs most frequently is ordered earlier.

Modified Depth-first Ordering (Scheme 2)[22,15]

The ‘don’t’ care fault tree is considered to be made up of sub-trees, each of which is fully explored in turn (from left to right as they appear in the fault tree) according to Scheme 1.

Modified Priority Depth-first Ordering (Scheme 3)[19]

Scheme 3 is a modified version of Scheme 2. Instead of considering gates from left to right as they appear in the ‘don’t care’ fault tree, scheme 3 considers gates with only event inputs before those with both event and gate inputs.

Modified Leaves Depth-first Ordering (Scheme 4)[22,19]

Scheme 4 is also a modified version of Scheme 2. A gate is considered first if it:

1. Contains the smallest number of leaves (total number of basic events beneath the gate). In case of ties, then

2. Contains the smallest number of unconsidered leaves.

Variables that occur most frequently appear earlier in the variable order list.

Non-Dynamic Top-down Weighted Ordering (Scheme 5) [11]

A variable in the 'don't care' fault tree is listed before others if it has:

1. The biggest contribution weight to the top event (the top event is given weight 1; the weight of each gate is then equally distributed between its inputs). Weights of repeated variables are added together. In case of ties, then
2. The smallest average level of appearance (the sum of the levels on which the variable appears, divided by how many times it occurs). In case of ties, then
3. The highest number of occurrences. In case of ties, then,
4. The highest priority in the order list when applying scheme 1.

Dynamic Top-down Weighted Ordering (Scheme 6) [21]

Scheme 5 is applied to a changing series of 'don't care' fault trees, each of which is created by deleting an allocated variable from the 'don't care' fault tree following its allocation. Weights are then reassigned to each modified fault tree to allocate another variable. The process repeats until all variables are allocated.

Bottom-up Weighted Ordering (Scheme 7) [22]

Variables under gates are sorted according to Scheme 1 with a gate being explored first if it has:

1. The highest weight. In case of ties,
2. The highest percentage of repeated variables (divide the total number of leaves by the number of repeated leaves).

A gate's weight is calculated according to the weight of its inputs:

1. The weight of an AND gate: $W_{AND} = \prod_{i=1}^{n} q_i$,
2. The weight of an OR gate: $W_{OR} = 1 - \prod_{i=1}^{n}(1 - q_i)$,

where q_i is the weight of the i^{th} input of the gate. Basic event inputs are given a weight of $q_i = 1/(m_{max} + 1)$, where m_{max} is the maximum number of failure modes related to any variable.

Event Criticality Ordering (Scheme 8)[23][24]

Variables in a 'don't care phase' fault tree (a mission failure fault tree where phase indices are neglected) are ordered according to a version of Birnbaum's structural importance measure, modified to account for multiple failure modes. Firstly, the importance measure for each component A in each of its failure mode i is calculated using:

$$I(A^i) = Q(1_{A^i}, \mathbf{q}) - Q(0_{A^i}, \mathbf{q}). \tag{12}$$

$Q(1_{A^i}, \mathbf{q})$ is the top event probability with probability 1 for event A^i, probability 0 for any event A^j $(i \neq j)$ and probability $q = 1/(m_{max} + 1)$ for any of the remaining events; $Q(0_{A^i}, \mathbf{q})$ is the top event probability with probability 0 for event A^i and probability $q = 1/(m_{max} + 1)$ for any of the remaining events; m_{max} is the maximal number of failure modes experienced by any component in the 'don't care phase' fault tree.

The Birnbaums structural importance measure for component A is then calculated by:

$$I(A) = \sum_{i=1}^{m_A} \frac{1}{m_A} * I(A^i), \tag{13}$$

where m_A is the number of failure modes in which component A can fail in the 'don't care phase' fault tree.

A component is ordered earlier if it:

1. Has the highest Birnbaums structural importance measure value. In case of ties;

2. Appears earlier in the "don' t' care phase' fault tree in a top-down, left-right manner.

For all of the schemes, fault tree basic event inputs are considered before gate inputs and if basic events or gates cannot be sorted using the principles of the schemes, those that appear earlier, i.e. towards the top left of the fault tree, are given priority.

Appendix B: DEP-BDD Model

All of the existing variable ordering schemes can be used to order variables in a PMS at a component level. The DEP-BDD model requires backward phase ordering and backward failure mode ordering and considers ordering at phase level before at failure mode level.

The DEP-BDD model[7] uses the following rules to compute the operation between two nodes $F = ite < x, F1, F0 >$ and $G = ite < y, G1, G0 >$. Supposing $x \leq y$, and $cp(x)$ and

$fm(x)$ are the component variable x relates to and the failure mode it fails in respectively, then: $F \diamond G =$

$$\begin{cases} ite < x, F1 \diamond G1, F0 \diamond G0 > & x = y \\ ite < x, F1 \diamond G, F0 \diamond L_0 > & cp(x) = cp(y), fm(x) = fm(y) \\ ite < x, F1 \diamond L_1, F0 \diamond G > & cp(x) = cp(y), fm(x) \neq fm(y) \\ ite < x, F1 \diamond G, F0 \diamond G > & x \neq y. \end{cases} \tag{14}$$

where $L_1 = (G0)_{x=1}$, $L_0 = (G0)_{x=0}$ is the first node with variable relating to a component other than x encountered during a traversal down the 0-branches of the BDD starting from G.

For a general node $G = ite < x, G1, G0 >$, where $G1 = ite < y, H1, H0 >$ and $G0 = ite < z, I1, I0 >$, the probability is calculated as follows[7]: $P(G) =$

$$\begin{cases} p(x) * P(G1) + [1 - p(x)] * P(G0) & \text{case 1} \\ P(G1) + P(G0) - P(H0) + p(x) * [P(K_1) - P(G0)] & \text{case 2} \\ P(G0) + p(x) * [P(G1) - P(K_2)] & \text{case 3} \\ P(G1) + P(G0) - P(H0) + p(x) * [P(K_1) - P(K_2)] & \text{case 4} \end{cases} \tag{15}$$

where K_1, and K_2 are the first node with variable relating to a different component to x encountered during the traversal down the 0-branch of node $G1$ and $G0$ respectively. The cases for the relationships between x, y and z are:

case 1: $cp(x) \neq cp(y)$ and $cp(x) \neq cp(z)$.

case 2: $cp(x) = cp(y)$, $fm(x) = fm(y)$ and $cp(x) \neq cp(z)$.

case 3: $cp(x) \neq cp(y)$ and $cp(x) = cp(z)$, $fm(x) \neq fm(z)$.

case 4: $cp(x) = cp(y)$ and $cp(x) = cp(z)$, $fm(x) = fm(y)$, $pn(x) \neq pn(y)$ and $fm(x) \neq fm(z)$.

References

1. Mo, Y., Han, J., Zhang, Z., Pan, Z. & Zhong, F. Approximate reliability rvaluation of large-scale distributed systems. *Journal of Information Science & Engineering* **30**, 15 (2014).

2. Andrews, J., Remenyte-Prescott, R. & Downes, C. Reliability Analysis in Responsive Mission Planning for Autonomous Vehicles. In *Proceedings of the 26th International System Safety Conference* (Vancouver, 2010).

3. Prescott, D. R., Remenyte-Prescott, R., Reed, S., Andrews, J. D. & Downes, C. A reliability analysis method using binary decision diagrams in phased mission planning. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability Analysis* **223**, 133–143 (2009).

4. Remenyte-Prescott, R., Andrews, J. D. & Chung, P. W. H. An efficient phased mission reliability analysis for autonomous vehicles. *Reliability Engineering and System Safety* **95**, 226–235 (2010).

5. Zang, X., Sun, H. & Trivedi, K. S. A BDD-based algorithm for reliability analysis of phased-mission systems. *IEEE Transactions on Computers* **48**, 50–60 (1999).

6. Poole, J. *A fast reliability analysis for unmanned aerial vehicles performing a phased mission*. Ph.D. thesis (2011).

7. Tang, Z. & Dugan, J. B. BDD-based reliability analysis of phased-mission systems with multimode failures. *IEEE Transactions on Reliability* **55**, 350–360 (2006).

8. Reed, S., Andrews, J. D. & Dunnett, S. J. Improved efficiency in the analysis of phased mission systems with multiple failure mode components. *IEEE Transactions on Reliability* **60**, 70–79 (2011).

9. Andrews, J. D., Poole, J. & Chen, W. H. Fast mission reliability prediction for Unmanned Aerial Vehicles. *Reliability Engineering and System Safety* **120**, 3–9 (2013). URL `http://www.sciencedirect.com/science/article/pii/S095183201300063X`.

10. Bryant, R. E. Gragh-based algorithms for Boolean funtion manipulation. *IEEE Transactions on Computers* **35**, 677–691 (1986).

11. Mo, Y. Variable ordering to improve BDD analysis of phased-mission systems with multimode failures. *IEEE Transactions on Reliability* **58**, 53–57 (2009).

12. Reed, S. *Methods for the efficient measurement of phased mission system reliability and component importance*. Ph.D. thesis, Loughborough university (2010).

13. Rauzy, A. New algorithms for fault tree analysis. *Reliability Engineering and System Safety* **40**, 203–211 (1993).

14. Prescott, D. R., Andrews, J. D. & Downes, C. Multi-platform phased mission reliability modelling for mission planning. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability Analysis* **223**, 27–39 (2009).

15. Mo, Y., Zhong, F., Liu, H., Yang, Q. & Cui, G. Efficient ordering heuristics in binary decision diagram-based fault tree analysis. *Quality and Reliability Engineering International* **29**, 307–315 (2013).

16. Mo, Y. New insights into the BDD-based reliability analysis of phased-mission systems. *IEEE Transactions on Reliability* **58**, 667–678 (2009).

17. Shaaban, E., Salem, A. & Moniem Wahdan, A. An interleaving based algorithm for ordering variables in shared BDDS. In *The 14th International Conference on Microelectronics*, 256–259 (IEEE, 2002).

18. Esary, J. D. & Proschan, F. Coherent structures of non-identical components. *Technometrics* **5**, 191–209 (1963). URL `http://www.jstor.org/stable/1266063`.

19. Mo, Y., Zhong, F., Zhao, X., Yang, Q. & Cui, G. New results to BDD truncation method for efficient top event probability calculation. *Nuclear Engineering and Technology* **44**, 755–766 (2012). URL `http://koreascience.or.kr/journal/view.jsp?kj=OJRHBJ&py=2012&vnc=v44n7&sp=755`.

20. Mo, Y., Xing, L. & Dugan, J. B. MDD-Based Method for Efficient Analysis on Phased-Mission Systems With Multimode Failures. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **44**, 757–769 (2014). URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6587608`.

21. Remenyte-Prescott, R. *System Failure Modelling Using Binary Decision Diagrams*. Ph.D. thesis (2007).

22. Bouissou, M., Bruyere, F. & Rauzy, A. BDD based fault-tree processing: a comparison of variable ordering heuristics. In *Proceedings of European Safety and Reliability Association Conference, ESREL'97* (1997).

23. Zhang, Y. *Development of BDD Models for Decision Support in Phased Mission Systems*. Ph.D. thesis, The University of Nottingham (2016).

24. Bartlett, L. M. & Andrews, J. D. Comparison of two new approaches to variable ordering for binary decision diagrams. *Quality and Reliability Engineering International* **17**, 151–158 (2001).

Biographies

Yang Zhang is a Research Associate in the Resilience Engineering Research Group at the University of Nottingham. She received the B.S degree in statistics from Shandong University, China, in 2010, the M.S. degree in mathematical finance from Loughborough University, UK, in 2011, and the Ph.D. degree in civil engineering from the University of Nottingham, UK, in 2016. Her Research interests focus on risk and reliability analysis for phased mission systems as a support tool for decision making and maintenance process modelling using different techniques.

Darren Prescott is Assistant Professor in Risk and Reliability Engineering in the Resilience Engineering Research Group at the University of Nottingham. His current research interests include aircraft fleet maintenance modelling, the application of reliability modelling techniques to support decision making in autonomous systems and the development of asset management models for the railway network. He has published around 50 papers in the areas of risk, reliability and maintainability. He is chair of the ESRA (European Safety and Reliability Association) Technical Committee on Aeronautics and Aerospace.