Minimal cutsets-based reduction approach for the use of binary decision diagrams on probabilistic safety assessment fault tree models

C Ibáñez-Llano[1]*, A Rauzy[2], E Meléndez[3], and **F Nieto[1]**
[1]Instituto de Investigación Tecnológica, Universidad Pontificia Comillas, Madrid, Spain
[2]Dassault Systèmes, Vélizy-Villacoublay, France
[3]Consejo de Seguridad Nuclear, Madrid, Spain

The manuscript was received on 19 May 2009 and was accepted after revision for publication on 2 September 2009.

DOI: 10.1243/1748006XJRR259

Abstract: Binary decision diagrams (BDDs) are a well-known alternative to the minimal cutsets (MCS) approach to assess Boolean reliability models. While the application of fault tree analysis can be considered to be consolidated, its application to the event trees involved in the probabilistic safety assessment (PSA) studies of the nuclear industry require extended efforts. For many real PSA models the full conversion procedure remains out of reach in terms of computational resources owing to their size, non-coherency, redundancy, and complexity. A potential solution to improve the quality of assessment methods is to design hybrid algorithms that combine the information derived from the calculation of MCS with the BDD methodology.

As a first step to develop this new approach, this paper explores various procedures and strategies based on this principle. First, a method is presented to reduce the fault tree model by considering only the domain of the most relevant MCS of the system prior to the BDD conversion and the impact on the final probability of the model is analysed. Second, several ordering heuristics derived from the MCS and the structural information of the model are proposed and compared, both in terms of their general performance and their sensitivity to the initial rewriting of the model. This preliminary study is applied on a set of fault tree models belonging to a real PSA study. The results obtained lead to some promising conclusions: it is shown that the topological information proves to be essential for the ordering and conversion procedures; it is also revealed that the rewriting strategies should be considered when designing variable ordering methods; and, finally, it is demonstrated that the reduction procedure provides a faster computation process without affecting the final probability. The long-term objective, which has motivated this work, is to apply this reduction procedure to quantify sequences of linked fault trees, both static and dynamic, a task for which further work is required.

Keywords: probabilistic safety assessment, fault tree analysis, binary cecision diagrams, minimal cutsets, variable ordering heuristics

1 INTRODUCTION

The fault tree/event tree methodology is widely used in the nuclear industry to obtain response models for probabilistic safety assessment (PSA) studies. The classical methodology to assess these models is based on the computation of the minimal cutsets (MCSs) and bound approaches. Owing to the complexity of the calculation and the large size of the models, truncation cut-offs on probability and also on the order of the MCS have to be applied to avoid ending up with too many MCSs.

Bryant's binary decision diagrams (BDDs [**1**]) are a well-known alternative to the MCS approach to

**Corresponding author: Instituto de Investigación Tecnológica, Universidad Pontificia Comillas, Alberto Aguilera 25, Madrid 28015, Spain.*
email: cristina.ibanez@iit.upcomillas.es

assess these models. Conversely to the classical methodology, the BDD approach involves no approximation in the quantification of the model and is able to handle correctly negative logic (success branches in the event trees). However, BDDs are also subject to combinatorial explosion because the BDD structure exponentially increases according to the number of variables. Moreover, the final size of the BDD is highly sensitive to the variable ordering needed to convert the Boolean model into a BDD.

After two decades of application, the BDD methodology has been devoted mainly to fault tree analysis, where it has been applied successfully to assess small and medium-sized fault tree models (typically with up to several hundreds of basic events). Recent works have also tackled its application to event tree assessment [**2**, **3**]. Attempts to apply it to very large models, however, such as the ones coming from PSA studies of the nuclear industry, which include several thousand basic events and logic gates, to the present date remains out of reach in terms of computational resources for many real cases. Although some attempts have been successful [**3**], for such large models it might not be possible to compute the BDD within reasonable amounts of time and computer memory without considering truncation or simplification of the model. Therefore, a potential solution to improve the quality of assessment methods is to develop a hybrid approach and to design algorithms and procedures that combine the calculation of MCS with the BDD approach. This is the primary motivation for the current work.

As a first step to develop this new hybrid methodology, the present paper presents a series of numerical tests designed to explore this novel approach concerning two important aspects in the application of the BDD technology to assess reliability Boolean models, which constitute the two main contributions of this paper. The first aspect is to explore the idea of reducing the model through the information provided by the set of the most relevant MCSs of the model, following the same principle as the classical approach. The impact of this approximation on the model quantification is evaluated. Second, with respect to the variable ordering problem, several ordering methods based on the information provided by the models are presented and tested, by comparing both its general performance and its sensitivity to the initial rewriting of the model.

The case study selected as a basis for the different numerical tests performed comes from a medium-sized event tree belonging to a real Spanish PSA study. It has 19 sequences, a total of 1649 basic events and nine functional events defining the branching points, from which six are defined by means of fault trees. The efficiency of this new hybrid approach is tested over this set of real fault trees as a preliminary step to validate its suitability to assess the full sequences of linked fault trees, both static and dynamic.

The remainder of the paper is organized to present the main contributions, as previously mentioned, in the first two sections. Section 2 reviews briefly the existing approaches for fault tree assessment and presents the results obtained for the first numerical test regarding the effect on the quantification obtained by applying the reduction to the model proposed in this hybrid approach. Section 3 is focused on the variable ordering topic. After a brief review of different approaches existing within the literature, the results obtained by applying several of these approaches are presented. Finally, section 4 presents the conclusions and future works.

2 HYBRID APPROACH TO ASSESS FAULT TREES

This section is devoted to a brief review of the two different existing approaches to assess Boolean models in the context of PSA studies, and to present the current proposed numerical approach to obtain a hybrid method relying upon a MCS-based reduction, which is described in the following sections.

2.1 MCS methodology

The majority of the assessment tools used for reliability analysis implement the classical methodology. This approach, broadly used and accepted, relies on the computation of MCS. For a presentation of the mathematical foundations of MCSs, the reader should refer to reference [**4**]. Classical algorithms to compute MCSs work either top-down or bottom-up (e.g. see references [**5**] and [**6**]). Owing to the complexity of the calculation and the large size of real models, several approximations have to be considered when applying this methodology.

First, the model is simplified. Algorithms to compute MCSs apply cutoffs on the probability and order of the MCS to avoid combinatorial explosion in the number of MCSs. In general, only a few thousand cutsets are eventually kept. The choice of the correct truncation value is the result of a trade-off between accuracy and computational time efforts and memory space requirements. Second, approximations in probability have to be considered. In order to assess probabilistic quantities from the MCS, either the rare event approximation, which corresponds to the first term of the Sylvester–Poincaré development to compute the probability of a union of events as

expressed in equation (1), or the min-cut upper bound approximations, are used.

$$p(E_1 \cup E_2 \cdots \cup E_k) = \sum_{1 \leqslant i \leqslant k} p(E_i) - \sum_{1 \leqslant i < j \leqslant k} p(E_i \cap E_j) + \cdots + 1^{-k} p(E_1 \cap \ldots \cap E_k) \quad (1)$$

This leads to a conservative approximation and therefore to an unknown error bound. Another major problem, which will not be considered here, is the fact that this approach is not able to deal with negations, which comes to be especially important when dealing with sequences of linked fault trees and success branches.

2.2 Binary decision diagrams

In order to overcome the limitations of the classical methodology, in the early 1990s Bryant's BDDs [**7**, **8**] were introduced into reliability analysis in order to have a more powerful tool to handle Boolean reliability models.

Binary decision diagrams are the state-of-the-art data structure to encode the truth tables of Boolean functions. Their representation is based on the Shannon decomposition of the function f defined in terms of the ternary connective *ite* (*if-then-else*), which allows the function to be divided into two disjoint components, both of which do not depend on the decision variable x

$$f = ite(x, f|_{x=1}, f|_{x=0}) = (x \wedge f|_{x=1}) \vee (\bar{x} \wedge f|_{x=0}) \quad (2)$$

By choosing a total order of all the variables and applying Shannon decomposition recursively, the truth table of the function can be represented by a binary tree. This Shannon tree is, in general, not compact, and contains redundant nodes and isomorphic subgraphs. It is, however, possible to apply reductions rules for these nodes and obtain the BDD associated with the formula.

A BDD is composed of terminal and non-terminal nodes connected by branches. Each internal or non-terminal node of the graph represents a variable of the function, and has two outgoing labelled branches. They indicate either occurrence or non-occurrence of the variable: left or *then* branch, labelled with 1, and represented by a solid line, points to the son node encoding $f(x=1)$, and right or *else* branch, labelled with 0, and represented by a dotted line, points to the son node encoding $f(x=0)$. All paths through the BDD start at the root vertex and terminate at one of the two terminal nodes, labelled 0 and 1, which represent the constant functions. Figure 1 shows an example of both Shannon tree and final reduced BDD of the function $f = x_1\, x_2 + x_3$. For details about those algorithms, the reader should refer to references [**9**] and [**10**]. For details about issues related to an efficient implementation of a BDD package, see reference [**1**].

Fault tree analysis is a two-fold problem: it involves both quantitative and qualitative aspects. The BDD approach has made it possible to improve the efficiency and the accuracy of both sides of analysis, allowing exact probabilistic quantification, compact encoding of the MCS, and correct handling of negative logic and non-coherent systems. In particular, one of its major advantages is that the probability of the top event can be obtained directly from the BDD, as a sum of probabilities of the disjoint paths. This is possible because paths through the BDD are mutually exclusive. Thus, as a result of the Shannon decomposition, the following equality can be applied to calculate the probability of failure $p(f)$ of a system f, where $p(x)$ stands for the probability of failure of a basic component, x

$$p(f) = p(x)\; p\left(f_{\{x=1\}}\right) + (1 - p(x))\; p\left(f_{\{x=0\}}\right) \quad (3)$$

This equality induces a recursive algorithm, which is linear in the size of the BDD and gives exact values [**10**].

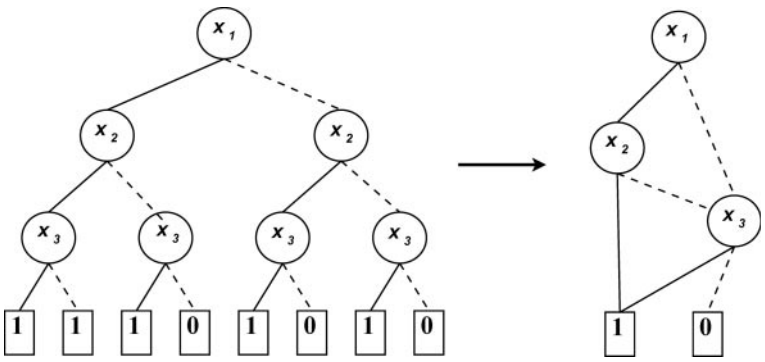

Fig. 1 An example of the Shannon tree and the final reduced BDD of the function $f = x_1\, x_2 + x_3$

2.3 Hybrid approach: reduction of the model

Despite all its good properties and the improvements that the BDD methodology offers to assess this kind of model, it might not be possible to convert the full model to the BDD when it consists of a large number of basic events and gates, especially if many of the events are repeated within the tree or sequence structure.

In such cases the problem needs to be reduced to a manageable size by discarding the less significant failure modes and retaining only the most relevant failure paths, following the same principle as the classical MCS approach does. This approximation is justified by the fact that these cutsets capture, in general, the most relevant part of the model in terms of the contribution to the top-event probability. Several works propose the computation of the BDD with some truncation limits to avoid the memory explosion, namely, the truncated BDD [**4**, **11**]. The idea is that, even if truncation in the number of MCSs is considered, the submodel derived from it could be assessed exactly and more efficiently by means of the BDD method.

An alternative procedure proposed in this work is to consider the reduction and simplification of the model before tackling the BDD conversion. Thus, first the MCSs with some cut-off value are computed by means of some classical algorithm, and then the fault tree model is reduced by deleting from the logic those variables and gates that are not involved in any significant MCS. Finally, with the new reduced fault tree model the BDD conversion and the probability quantification can be performed. Since the number of variables and gates appearing in the reduced model is significantly lower than in the full model, the BDD conversion can be more easily achieved. The issue which needs to be studied is how much this reduction affects the quantification result. To illustrate the procedure, consider the following example of a simple fault tree model represented in Fig. 2 and defined by the following set of equations

$$
\begin{aligned}
r_1 &= g_1 \wedge g_2 \\
g_1 &= g_3 \vee e_1 \vee e_2 \vee e_3 \\
g_2 &= e_4 \vee e_5 \vee e_6 \\
g_3 &= g_4 \wedge g_5 \\
g_4 &= e_7 \vee e_8 \\
g_5 &= g_6 \vee e_9 \vee e_{10} \vee e_{11} \\
g_6 &= e_1 \wedge e_9 \wedge e_{12}
\end{aligned}
$$

This example will be used throughout the paper to describe the suggested procedures and methods. The set of formulae are rooted by an event called the top event (in Fig. 2 it corresponds to r_1). This system has a

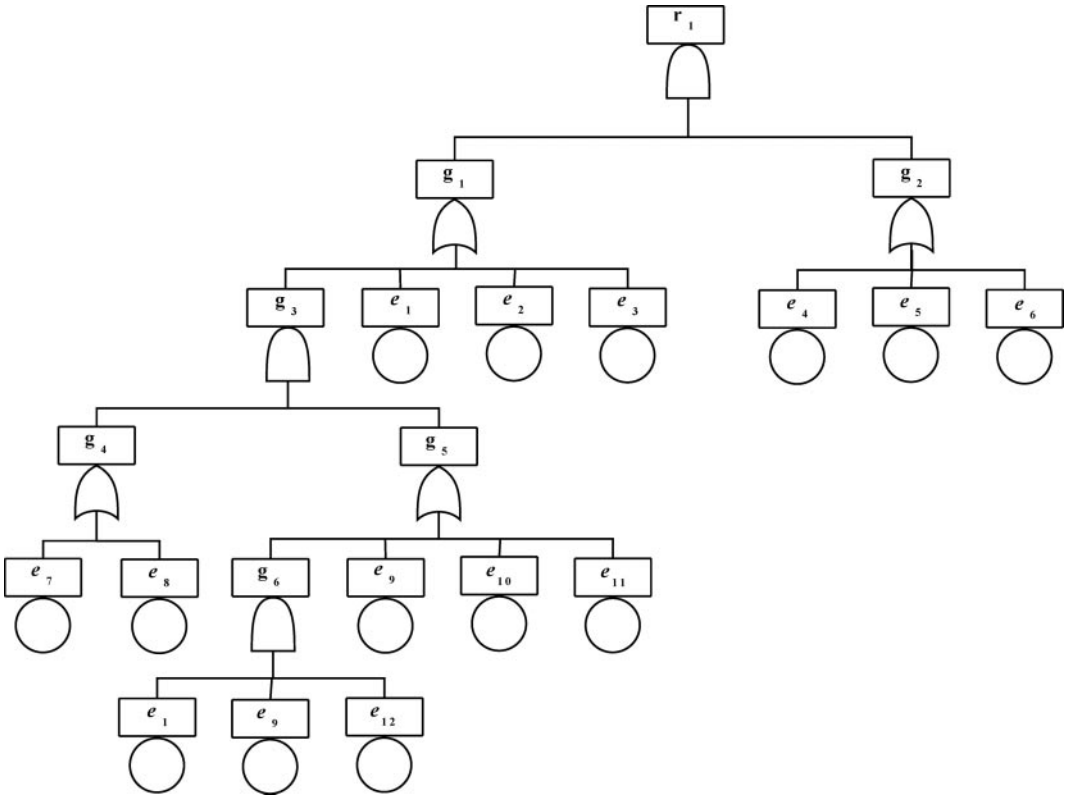

Fig. 2 A simple fault tree model

total number of six gates, 12 basic events or variables, and 24 MCSs

$\{e_2, e_4\}$	$\{e_1, e_4\}$
$\{e_2, e_5\}$	$\{e_1, e_5\}$
$\{e_2, e_6\}$	$\{e_1, e_6\}$
$\{e_7, e_9, e_4\}$	$\{e_8, e_9, e_4\}$
$\{e_7, e_9, e_5\}$	$\{e_8, e_9, e_5\}$
$\{e_7, e_9, e_6\}$	$\{e_8, e_9, e_6\}$
$\{e_7, e_{10}, e_4\}$	$\{e_8, e_{10}, e_4\}$
$\{e_7, e_{10}, e_5\}$	$\{e_8, e_{10}, e_5\}$
$\{e_7, e_{10}, e_6\}$	$\{e_8, e_{10}, e_6\}$
$\{e_7, e_{11}, e_4\}$	$\{e_8, e_{11}, e_4\}$
$\{e_7, e_{11}, e_5\}$	$\{e_8, e_{11}, e_5\}$
$\{e_7, e_{11}, e_6\}$	$\{e_8, e_{11}, e_6\}$

Now, suppose that the 12 rightmost MCSs are discarded for having a probability lower than a given cut-off value. From the remaining set of MCSs, the variables which are missing are e_1 and e_8, so they have to be discarded from the fault tree model. This implies eliminating gate g_6 as well. The resulting reduced model, having now ten basic events, is shown in Fig. 3.

2.4 Model reduction: numerical results

The purpose of this section is to compare the original model against the reduced model in terms of the size of the model as well as in terms of the top event probability. Concerning this last comparison, another goal is also to compare the results obtained with the BDD methodology with those of MCS-based calculations.

In order to test the reduction procedure, the MCSs of the six fault trees of the case study (F1 to F6) were obtained by means of classical algorithms with two different cut-off values. The cut-off used in practice for the analysis of the event tree is around 10^{-12}/10^{-14}. Considering that this test relates to each of the fault trees defining its branching points, the values selected to analyse these models were established in 10^{-10} and 10^{-11}. These MCSs were used to obtain the discarded variables in order to perform the reduction procedure of each model. Table 1 presents the basic statistics regarding the number of basic events and

Table 1 Statistics of the fault trees for the complete and reduced models of the case study ('red.' denotes 'reduced')

Fault tree	Cut-off reduction	No. of basic events	No. of MCS
F1	Complete	275	45 055
	Red. (10^{-11})	241	698
	Red. (10^{-10})	202	418
F2	Complete	797	2.96×10^{12}
	Red. (10^{-11})	577	94 111
	Red. (10^{-10})	539	41 980
F3	Complete	767	6.10×10^{12}
	Red. (10^{-11})	547	41 469
	Red. (10^{-10})	420	16 861
F4	Complete	731	3.21×10^{12}
	Red. (10^{-11})	461	20 975
	Red. (10^{-10})	376	11 236
F5	Complete	79	534
	Red. (10^{-11})	79	312
	Red. (10^{-10})	79	249
F6	Complete	657	5.29×10^{13}
	Red. (10^{-11})	236	4126
	Red. (10^{-10})	161	891

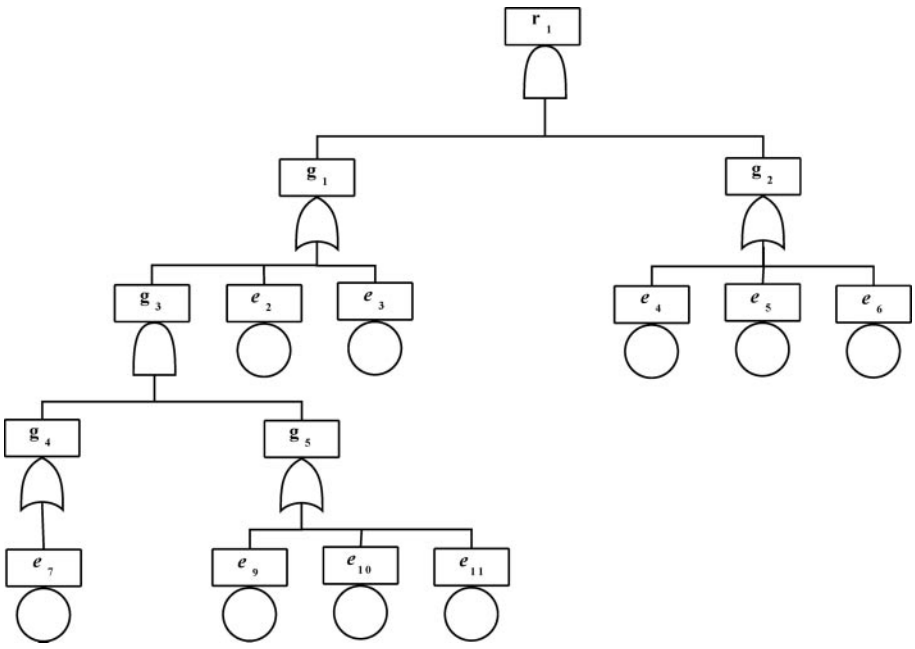

Fig. 3 The reduced model after deleting variables e_1, and e_8, and gate g_6

MCSs of each example, for both the full fault tree model and the reduced models, as a measure to describe its complexity. It can be seen that the bigger the model is, the larger the reduction is, in terms of the number of basic events considered as relevant in the model. Table 2 presents the top event probabilities computed with a classical tool [**12**] using the MCS approach and the rare event approximation, and the BDD approach. Notice that in the case of fault tree F5, even if the number of MCSs decreases when truncation is considered (fourth column of Table 1), there are no discarded variables in the set of relevant MCSs with any of the reductions (third column of Table 1). This is because the model is simple enough to keep all the variables when computing the MCSs, so the reduction was not performed.

Regarding the comparison of the classical and the BDD approaches (third and fourth columns of Table 2 respectively) the differences can be seen to be very small, which confirms the validity of the classical methodology. Additionally, when comparing the results obtained with the BDDs for the full and the reduced model, it can seen that the differences are even smaller than the previous ones, almost negligible, despite the significant decrease of the number of variables involved in the reduced model in relation with the total number of the complete model as shown in Table 1. This means that the quantification in terms of BDD of the reduced model gives more exact results than the ones computed directly from the MCSs. Moreover, as the model has been previously reduced, the conversion procedure to the BDD is very fast and does not suffer from combinatorial explosion.

Table 2 Top event probabilities computed with the MCS approach and the BDD approach

		Top probability	
Fault tree	Cut-off reduction	MCS approach	BDD approach
F1	Complete	–	$3.577\,745\times10^{-2}$
	Red. (10^{-11})	3.5810×10^{-2}	$3.577\,745\times10^{-2}$
	Red. (10^{-10})	3.5810×10^{-2}	$3.577\,744\times10^{-2}$
F2	Complete	–	$3.816\,642\times10^{-3}$
	Red. (10^{-11})	3.9370×10^{-3}	$3.816\,639\times10^{-3}$
	Red. (10^{-10})	3.9370×10^{-3}	$3.816\,613\times10^{-3}$
F3	Complete	–	$2.336\,591\times10^{-3}$
	Red. (10^{-11})	2.3610×10^{-3}	$2.336\,589\times10^{-3}$
	Red. (10^{-10})	2.3610×10^{-3}	$2.336\,510\times10^{-3}$
F4	Complete	–	$2.850\,081\times10^{-3}$
	Red. (10^{-11})	2.8650×10^{-3}	$2.850\,080\times10^{-3}$
	Red. (10^{-10})	2.8650×10^{-3}	$2.850\,074\times10^{-3}$
F5	Complete	–	$1.819\,709\times10^{-2}$
	Red. (10^{-11})	1.8200×10^{-2}	$1.819\,709\times10^{-2}$
	Red. (10^{-10})	1.8200×10^{-2}	$1.819\,709\times10^{-2}$
F6	Complete	–	$2.436\,747\times10^{-6}$
	Red. (10^{-11})	2.4480×10^{-6}	$2.431\,856\times10^{-6}$
	Red. (10^{-10})	2.4220×10^{-6}	$2.415\,812\times10^{-6}$

3 VARIABLE ORDERING STRATEGIES

It is well known that the final size of the BDD, and therefore the efficiency of the whole methodology, depends heavily on the chosen variable ordering. Already in the mid-1980s Bryant addressed the need for a good ordering so as to consolidate the technology [**7**]. Finding the optimal ordering is a NP-hard (NP referring to non-deterministic polynomial time) problem [**13**] owing to the combinatorial nature, so it is computationally intractable [**14**]. In order to find reasonably good orderings, research efforts have been aimed at the design of suitable heuristic methods as well as good preprocessing strategies (modularization, rewritings, simplifications, etc.).

3.1 Review of existing approaches

Variable ordering heuristic methods can be classified into two main groups: static heuristics and dynamic heuristics. The former produce an initial variable ordering prior to the BDD conversion and are based on topological considerations. The latter, on the other hand, are used to change variable ordering during the BDD conversion process, and are based on swapping and shifting groups of variables at some points of the computation [**15**]. Dynamic reordering is highly time-consuming. Thus, even if such methods may help to improve the final result, it is worthwhile to start with sufficiently good orderings given by some static heuristic.

Static methods can also be divided into different categories and approaches, but all of them are based on extracting some structural information of the formula under study (weights, redundancy, size of the subformulae, etc.). Some of them are specially designed for specific types of Boolean formulae, as, for example, the sum-of-products formulae, where the formula is basically a conjunction of disjunctions (similarly products of sums or disjunction of conjunctions).

Nevertheless, most of the proposed methods within the literature fall into one of the following two categories: structural methods or weighted methods. Structural methods [**16–19**] are based on performing a depth first traversal of the graph underlying the formulae, possibly after some rearrangement of the connective arguments. In practice, these heuristics give rather good results because they tend to preserve the structural locality of variables: variables that are close in the formula tend to be not far in the ordering. As they have not been outperformed by any other

method proposed in the literature they are usually taken as a comparison basis. The main drawback of these methods, however, is that they are not at all robust, as the final result is very sensible to the way the formula is written, as pointed out in previous studies [**17**, **20**].

Weighted methods [**21**–**23**] assign different measures to each variable, leading to a complete rearrangement of the whole list of variables. In contrast with the structural ones, these methods do not necessarily preserve the structural locality of the variables, although they are able to reduce the instability with respect to the rewritings. There are many heuristics proposed in the literature in this category, especially for electronic circuits. They are not applied in the reliability field, however, as they have not offered good performance for fault tree formulae.

In the reliability engineering framework, the disjunctive normal form (DNF) of the Boolean expression, also called the sum-of-products form, is of special interest because it represents the model expressed in terms of the MCS. In that case, each product π_i corresponds to a MCS and is made up of a conjunction of literals or basic events x_i as expressed in the following equation

$$\sum_{i=1}^{n} \pi_i = \sum_{i=1}^{n} \left(\prod_{j=1}^{m} x_i^j \right) \tag{4}$$

The idea that underlies this new approach is to investigate the potential of designing hybrid algorithms using the information given by the MCS, so the authors have investigated in a previous work [**24**] the use of the MCS to define different types of weighted methods to be applied specifically over sums-of-products instead of over the initial tree expression. The objective was to study whether the concept of connection between variables could be advantageously defined as the participation in the same MCS. Before explaining the methods, some preliminary definitions will be introduced.

Let S be a sum-of-products. A hypergraph $HG(S)$ induced by this set of products can be derived where the variables represent the vertices and the hyperedges represent the products. By creating an edge between all pairs of variables occurring in the same product, the hypergraph can always be transformed to a simple graph $G(S)$. Given S and $G(S)$, several measures can be defined for products or variables using the information given by $G(S)$ as follows.

First, the weight of a product is defined as a function, f, which should be inversely proportional to its size

$$w(\pi) = f(|\pi|) \tag{5}$$

Second, the weight of a variable is defined as the sum of weights of all products containing the variable

$$w(v) = \sum_{\pi : v \in \pi} w(\pi) \tag{6}$$

With these two definitions the weight of an ordering can be derived with the intent of measuring the strength of this ordering with respect to the importance of the connections defined by the edges. The aim of these definitions is to design heuristic methods close to those proposed in reference [**25**], which pursue the following principles:

(a) variables appearing in the same product should be close in the ordering;
(b) smaller products should have the priority.

Thus, the method proposed for the special case of models expressed in terms of their MCSs, the so-called MCS-weighted method (MCS-W) consists of assigning weights to products and variables according to definitions (5) and (6) respectively, and selecting variables one by one. The propagation of the weights is dynamic because each time a variable is selected it is removed from all products containing it, and weights are updated. The intention is to complete products and therefore to order consecutively variables occurring in the same cutset. The weight-based mechanism to choose a variable can be defined in several ways. In this paper two different mechanisms have been used to define different methods of this group:

(a) MCS-W1 selects first the shortest product and, within this product, selects the most frequent variable;
(b) MCS-W2 selects directly the variable with the highest weight.

The remainder of this section will be devoted to presenting the result of different numerical tests performed with several ordering heuristics: the basic structural heuristic, the depth-first transversal, and these MCS-weighted methods. The former are applied over the tree structure, whereas the latter are applied over the transformation of the model in terms of its MCS. As the objective is to measure the potential of the methods and to compare them with each other, there were no limitations on running time or memory space, and no special tuning of the BDD tables was applied. All cases were run in a Pentium V computer at 3.0 GHz and 1GB RAM.

3.2 Structural methods: sensitivity to initial formula writing

In this first test the BDD size was computed directly from the tree structure and with the basic depth-first heuristic, which is usually taken as a basis for comparison. As said previously, this heuristic depends on the way the formulae are written initially. Therefore,

the purpose of the test was to study the sensitivity to the initial formula writing. For that reason, the objective was to obtain the distribution of the BDD size for different initial rewritings of the tree. To achieve this, the following procedure was repeated 100 times.

1. *Step 1* Random rewriting of the tree. First a unique index is associated with each gate and basic event, and then these indices are randomly shuffled. Finally, the inputs of each gate are sorted according to these new indices. To illustrate this procedure consider the example already introduced in section 2.3 and represented in Fig. 2. All elements of the tree (except the top event) $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, g_1, g_2, g_3, g_4, g_5, g_6\}$ are assigned an index from 1 to 18 which are randomly permuted, leading for instance to the following set of new indices {6, 3, 16, 11, 7, 17, 14, 8, 5, 15, 1, 2, 4, 18, 13, 9, 10, 12}. Now, for instance gate g_1 is rewritten with the inputs in the following order: e_2, e_1, g_3, e_6. The complete tree rewritten using this permutation is shown Fig. 4.
2. *Step 2* Variable ordering. Once the tree is rewritten, the depth-first heuristic is applied to obtain the variable ordering. In the authors' current example this would be: $e_2 < e_1 < e_8 < e_7 < e_{11} < e_9 < e_{12} < e_{10} < e_3 < e_5 < e_4 < e_6$.
3. *Step 3*. Finally the BDD is computed using the previous ordering and the final size is recorded.

Table 3 shows the minimum, maximum, and mean value of the BDD sizes for the 100 runs obtained in

Table 3 BDD sizes of the fault trees for the complete and the reduced models with the depth-first heuristic

		|BDD| with depth-first method		
Fault tree	Cut-off reduction	Minimum	Relative mean	Relative maximum
F1	Complete	1623	2.78	7.26
	Red. (10^{-11})	850	3.16	7.34
	Red. (10^{-10})	731	2.32	5.91
F2	Complete	200 829	3.69	28.19
	Red. (10^{-11})	96 201	2.93	9.63
	Red. (10^{-10})	31 860	2.74	10.24
F3	Complete	344 889	2.91	10.53
	Red. (10^{-11})	207 150	2.25	6.67
	Red. (10^{-10})	24 438	2.21	5.76
F4	Complete	123 225	1.90	3.39
	Red. (10^{-11})	32 009	2.07	3.75
	Red. (10^{-10})	11 704	1.89	3.29
F5	Complete	125	1.69	2.59
	Red. (10^{-11})	–	–	–
	Red. (10^{-10})	–	–	–
F6	Complete	151 291	4.85	30.12
	Red. (10^{-11})	5151	2.20	4.89
	Red. (10^{-10})	1187	1.67	2.68

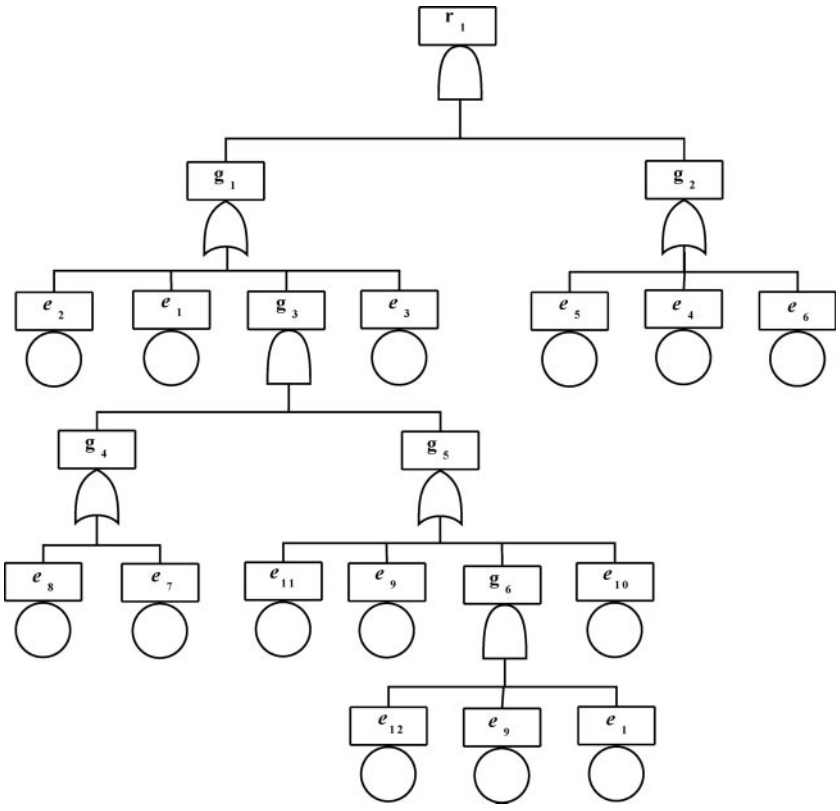

Fig. 4 The tree model of Fig. 2 after random rewriting

the test for each of the fault trees (F1 to F6) for both the complete model and the reduced models.

Recall that the BDDs were obtained directly from the tree structure. In order to compare the deviation between the different results the maximum and the mean values were divided by the minimum value to obtain the relative sizes. Results clearly prove the lack of robustness of this structural method against random rewritings, with factors up to 30 between the maximum and the minimum values in some cases. Thus, results confirm that the pure depth-first heuristic and in general the structural methods cannot be applied without taking this issue into account. Therefore, rewriting heuristics needs to be considered as part of the ordering methods, as elements of more adaptable and complex strategies.

Additionally, by comparing the results of the complete and reduced models for each case, a very important decrease in the BDD size can be observed for both reduced models (in some cases the reduction goes up to 90 per cent). This proves what was previously mentioned: the conversion procedure to the BDD from the reduced model is much more efficient owing to the reduction of the model complexity, so there exists a higher chance of obtaining the BDD.

3.3 Weighted methods: performance and comparison of MCS-based methods

In a second group of numerical tests, the objective was to test and compare the performance of different methods applied in this instance over the MCS obtained for each model, instead of over the tree. For each submodel derived from the MCS computation, the two variants of the MCS-weighted methods already exposed in section 3.1 were applied using three different measures of the weight of a cutset.

$$w_1(\pi) = 1,\ w_2(\pi) = 1/|\pi|,\ w_3(\pi) = \mathrm{e}^{-|\pi|}$$

Using w_1 implies that no priority is given to small products, as occurs with the basic depth-first heuristic, whereas w_2 and w_3 are intended to explore ways to give priority to small products more or less pronouncedly. Again, because these kinds of methods are not robust against rewritings, the sum-of-products representation of the formula was randomly rewritten to analyse the BDD size distribution obtained. Thus, from each example expressed in terms of MCS 100 different rewritings were derived. In the same way as with the tree structure, variables of the whole set of products were rearranged at random as follows. First, a unique index was associated at random with each variable. Then, variables inside products were sorted according to these indices. Finally, products were sorted inside the sum according to the indices following the lexicographical order.

The results of these methods are presented in Tables 4 and 5 respectively. Table 4 shows the minimum, maximum, and mean BDD sizes obtained with 100 different rewritings for the MCS-W1 method, whereas Table 5 shows a unique value as the method MCS-W2 is independent of the rewriting. In order to compare the results obtained from the MCS representation with those obtained directly from the tree on the same basis, all values have been normalized by the minimum value obtained with the depth-first heuristic applied directly to the tree structure of the correspondent reduced model (third column in Table 3).

Concerning the comparison of both tables the results show that the first variant proposed, MCS-W1, is in general better than MCS-W2, although it is

Table 4 BDD sizes of the MCS reduced models with the MCS-weighted W1 method

		|BDD| with MCS-W1 method								
		w_1			w_2			w_3		
Fault tree	Cut-off reduction	Min.	Mean	Max.	Min.	Mean	Max.	Min.	Mean	Max.
F1	Red. (10^{-10})	28.18	1036.79	5649.93	3.48	14.35	36.75	3.44	8.43	17.98
	Red. (10^{-11})	21.49	285.20	1703.53	3.30	8.69	19.18	3.43	8.42	17.37
F2	Red. (10^{-10})	OM*	OM	OM	OM	OM	OM	OM	OM	OM
	Red. (10^{-11})	OM	OM	OM	OM	OM	OM	OM	OM	OM
F3	Red. (10^{-10})	OM	OM	OM	OM	OM	OM	OM	OM	OM
	Red. (10^{-11})	11.24	234.76	1178.00	1.22	8.73	30.91	0.83	10.02	47.44
F4	Red. (10^{-10})	OM	OM	OM	2.50	15.97	76.15	2.84	18.90	156.26
	Red. (10^{-11})	17.85	893.51	2745.75	0.44	2.56	12.63	0.36	2.41	10.09
F5	Red. (10^{-10})	1.84	4.05	11.88	1.44	1.64	2.36	1.41	1.65	2.30
	Red. (10^{-11})	1.82	4.18	10.67	1.42	1.73	2.73	1.43	1.74	2.51
F6	Red. (10^{-10})	4.92	85.28	312.70	1.23	4.68	10.16	1.23	5.35	17.92
	Red. (10^{-11})	3.28	9.89	20.84	2.07	5.97	12.79	1.79	6.29	14.14

*OM denotes BDD computation out of memory.

Table 5 BDD sizes of the MCS reduced models with the MCS-Weighted W2 Method

		|BDD| with MCS-W2 method		
Model name	Type of model	w_1	w_2	w_3
F1	Red. (10^{-10})	10 950.52	2801.66	923.57
	Red. (10^{-11})	7513.41	3245.21	941.06
F2	Red. (10^{-10})	OM	OM	OM
	Red. (10^{-11})	OM	OM	OM
F3	Red. (10^{-10})	OM	OM	OM
	Red. (10^{-11})	OM	366.20	181.91
F4	Red. (10^{-10})	OM	OM	OM
	Red. (10^{-11})	OM	OM	2092.32
F5	Red. (10^{-10})	2.03	2.46	2.63
	Red. (10^{-11})	2.55	2.87	2.93
F6	Red. (10^{-10})	237.31	130.29	95.27
	Red. (10^{-11})	17.53	10.55	9.57

OM: BDD computation out of memory

difficult to compare a complete distribution against a unique execution.

Analysing the results of the MCS-W1 method, it can seen that using w_1 presents very bad results, whereas the variants which include w_2 and w_3 attain better results (several factors corresponding to the relative minimum value are close to 1). This corroborates the idea that it is beneficial to start ordering the smallest MCS rather than any of the whole set. Despite this favourable fact, however, the results clearly show that the weighted methods proposed for the special case of the MCS representation of the model are in general worse than those obtained using the structural information of the tree. Two additional points have to be considered. First, for a considerable number of cases it was not possible to compute the BDD, as the cases ran out of memory. Second, it has to be pointed out that, although the ordering heuristics are performed over the MCS representation of the model, the computation of the BDD for the reduced models was done from the simplified tree structure rather than from the MCS, because this latter approach is much more expensive in terms of computational time. In other words, this means that the topological information from the tree structure is an essential piece of information that has to be taken into account not only for the ordering, but for the conversion process as well.

4 CONCLUSIONS

In this paper a new approach has been presented for the assessment of PSA models by means of BDD. It is based on a procedure to reduce the fault tree model prior to the BDD conversion through the information given by the most relevant MCSs of the model. Additionally, several ordering methods based on the idea of grouping variables that appear in the same MCS were exposed. To assess and compare these methods, several tests were performed over a set of fault trees of a real PSA model.

From the set of tests presented the following conclusions can be obtained. The reduction procedure allows the memory explosion to be controlled by considering only the domain of the most relevant MCSs of the model. Thus, there is a higher chance of obtaining the BDD. Furthermore, from the initial test, it was shown that the probability quantification is almost unaffected by this reduction of the model in the case of the isolated fault tree models of the present case study, as the variables that are deleted are those which occur in the less probable failure cutsets. The advantage of this approach, however, is that it is not required to eliminate all the variables from the tree structure, only enough to be able to construct the BDD. Moreover, once the tree is reduced up to a certain point, the BDD conversion is much more easily computed, achieving a significant reduction of the BDD size. Hence, this new hybrid approach offers a promising and feasible procedure to assess sequences of event trees using the reduced BDDs, overcoming both the difficulties of the memory explosion (as they are large and complex models) and the correct treatment of the negative logic (as they include success branches).

Concerning the variable ordering heuristic, the results have clearly shown that the orderings derived from the MCS are not useful and that it is essential to preserve the tree topology for both the ordering and the conversion procedures, which reinforces the idea that the model has to be reduced before the BDD conversion, as proposed in the current work. Another important conclusion, which has been confirmed by the results, is that the existing methods are very sensitive to the way the formulae are written; this means that the heuristics should not be considered in isolation and that the rewriting procedures need to be included as part of the ordering strategies, as has already been pointed out in previous studies.

Future development and extensions of PSA studies are taking into account the dynamic interaction with the accident scenario evolution in the framework of the dynamic reliability methodologies. In the integrated sequence analysis methodology [**26**], the interaction with the dynamic evolution of the accident allows the generation by simulation of the dynamic event tree [**27**]. Its delineation implies devising an incremental algorithm that accumulatively builds and quantifies the sequences at each branching point. The next important step to develop this new hybrid approach fully is to extend it to the full sequences assessment and to integrate it with the incremental procedure required for dynamic event trees quantification, which is the long-term goal of the present authors' work. The authors are working in

this direction and the results for this further study will be presented in a future publication.

REFERENCES

1 **Brace, K., Rudell, R.,** and **Bryant, R. E.** Efficient implementation of a BDD package. In Proceedings of ACM/IEEE Design Automation Conference DAC'90, Orlando, Florida, 24–27 June 1990, pp. 40–45.

2 **Andrews, J. D.** and **Dunnett, S. J.** Event-tree analysis using binary decision diagrams. *IEEE Trans. Reliability*, 2000, **49**(2), 230–239.

3 **Epstein, S.** and **Rauzy, A.** Can we trust PRA? *Reliability Engng System Saf.*, 2005, **88**(3), 195–205.

4 **Rauzy, A.** Mathematical foundations of minimal cutsets. *IEEE Trans. Reliability*, 2001, **50**(4), 389–396.

5 **Fussel, J. B.** and **Vesely, W. E.** A new methodology for obtaining cut sets for fault trees. *Trans. Am. Nucl. Soc.*, 1972, **15**, 262–263.

6 **Jung, W. S., Han, S. H.,** and **Ha, J.** A fast BDD algorithm for large coherent fault trees analysis. *Reliability Engng System Saf.*, 2004, **83**, 369–374.

7 **Bryant, R. E.** Graph based algorithms for Boolean function manipulation. *IEEE Trans. Computers*, 1986, **35**(8), 677–691.

8 **Bryant, R. E.** Symbolic Boolean manipulation with ordered binary decision diagrams. *ACM Comput. Surv.*, 1992, **24**, 293–318.

9 **Rauzy, A.** BDD for reliability studies. In *Handbook of performability engineering* (Ed. K. B. Misra), 2008, pp. 381–396 (Springer).

10 **Rauzy, A.** A brief introduction to binary decision diagrams. *Eur. J. Automn*, 1996, **30**, 1033–1050.

11 **Rauzy, A., Châtelet, E., Dutuit, Y.,** and **Bérenguer, C.** A practical comparison of methods to assess sum-of-products. *Reliability Engng System Saf.*, 2003, **79**, 33–42.

12 Risk Spectrum Professional. Relcon AB, 1998.

13 **Bollig, B.** and **Wegener, I.** Improving the variable ordering of OBDD is NP-complete. *IEEE Trans. Computers*, 1996, **45**(9), 993–1001.

14 **Friedman, S. J.** and **Supowit, K. J.** Finding the optimal variable ordering for binary decision diagrams. *IEEE Trans. Computers*, 1990, **39**(5), 710–713.

15 **Rudell, R.** Dynamic variable ordering for ordered binary decision diagrams. In Proceedings of IEEE International Conference on *Computer aided design, ICCAD'93*, Santa Clara, CA, USA, 7–11 November 1993, pp. 42–47.

16 **Ibañez-Llano, C., Meléndez, E.,** and **Nieto, F.** Variable ordering schemes to apply to the binary decision diagram methodology for event tree sequences assessment. *Proc. IMechE, Part O: J. Risk and Reliability*, 2008, **222**(O1), 7–16. DOI: 10.1243/1748006XJRR67.

17 **Bouissou, M., Bruyere, F.,** and **Rauzy, A.** BDD based fault tree processing: a comparison of variable ordering heuristics. In Proceedings of European Safety and Reliability Association Conference, ESREL'97, Lisbon, Portugal, 17–20 June 1997, vol. 3, pp. 2045–2052.

18 **Sinnamon, R. M.** and **Andrews, J. D.** Improving efficiency in qualitative fault tree analysis. *Qual. Reliability Engng Int.*, 1997, **13**, 293–298.

19 **Bartlett, L. M.** *Variable ordering heuristics for binary decision diagrams*. PhD Thesis, University of Loughborough, UK, 2000.

20 **Nikolskaia, M.** and **Rauzy, A.** Fine-tuning of Boolean formulae preprocessing techniques. Proceedings of European Safety and Reliability Association Conference, ESREL'99, Munich, Germany, 13–17 September 1999, vol. 2, pp. 1027–1032.

21 **Fujita, M., Fujisawa, H.,** and **Matsugana, Y.** Variable ordering algorithm for ordered binary decision diagrams and their evaluation. *IEEE Trans. Computer-Aided Des. Integrated Circuits and Systems*, 1993, **12**(1), 6–12.

22 **Minato, S., Ishiura, N.,** and **Yajima, S.** Shared binary decision diagrams with attributed edges for efficient Boolean function manipulation. In Proceedings of the ACM/IEEE Design Automation Conference, Orlando, Florida, USA, 24–28 June 1990, pp. 52–57.

23 **Aloul, F. A., Markov, I. L.,** and **Sakallah, K. A.** FORCE: A fast and easy-to-implement variable-ordering heuristic. In Proceedings of the Great Lakes Symposium on VLSI (GLSVLSI), Washington DC, 28–29 April 2003, pp. 116–119.

24 **Ibañez-Llano, C.** and **Rauzy, A.** Variable ordering heuristics for BDD based on minimal cutsets. In Proceedings of the International Probabilistic Safety Assessment and Management Conference (PSAM 9) Hong Kong, 18–23 May 2008.

25 **Rauzy, A., Gauthier, J.,** and **Leduc, X.** Assessment of large automatically generated fault trees by means of binary decision diagrams. *Proc. IMechE, Part O: J. Risk and Reliability*, 2007, **221**(O2), 95–105. DOI: 10.1243/1748006XJRR47.

26 **Izquierdo, J. M., Hortal, J., Sánchez Perea, M.,** and **Meléndez, E.** *An integrated PSA approach to independent regulatory evaluations of nuclear safety assessments of Spanish nuclear power stations*. CSN Publication ODE-04.18, 2002 (CSN (Spanish Nuclear Safety Council), Spain).

27 **Izquierdo, J. M., Meléndez, E.,** and **Devooght, J.** Relationships between probabilistic dynamics and event trees. *Reliability Engng System Saf.*, 1996, **52**, 197–209.