MCS BDD – DESCRIPTION AND VERIFICATION OF THE METHOD IMPLEMENTED IN RISKSPECTRUM

Ola Bäckström[1], Pavel Krcal[1], Wei Wang[1]
[1] Lloyd's Register: P.O. Box 1288, Sundbyberg, Sweden, 17225, and ola.backstrom@lr.org

The classic Minimal Cut Set (MCS) generation method where minimum cut upper bound or 1st, 2nd or 3rd order approximation is used, is well suited for MCS with low probability events. The complexity of modern PSA models is constantly increasing. The complexity is driven by refined modelling of dependencies and system behavior, but also the inclusion and more realistic modelling of external events. To handle the increasing need for accuracy a Binary Decision Diagram (BDD) calculation method based on the MCS list has been developed as a part of the RiskSpectrum package.

The advantage with the RiskSpectrum MCS BDD algorithm is both the accuracy in the quantification of high probability events and the calculation of success within a sequence. With the combination of exact and approximate treatment, the algorithm is scalable and efficient. This paper describes the method applied and also describes the verification approach for the algorithm on a high level.

I. INTRODUCTION

The classic Minimal Cut Set (MCS) generation method where minimum cut upper bound or 1st, 2nd or 3rd order approximation is used, is well suited for MCS with low probability events. However with an increasing number of high probability events in today's PSAs an alternative Binary Decision Diagram (BDD) calculation method based on the MCS list has been developed as a part of the RiskSpectrum package.

Event trees and fault trees encode huge Boolean formulae with probabilities assigned to the atomic variables. Boolean Decision Diagrams were proposed as an efficient data structure for manipulations with Boolean formulae (Ref. 1) and have been applied in various areas. They were introduced in the field of reliability as an alternative to the cutset based approach to solve the event trees and fault trees (Refs. 2 and 3). Despite of numerous efforts (Ref. 4), it has not been possible to solve real-life PSA models with a BDD-based technology. Instead, hybrid methods using cutsets and BDDs, such as quantification of a MCS list by the means of a BDD, have been proposed. This paper reports about specific aspects of the BDD quantification of MCS lists (*MCS BDD*) implemented in RiskSpectrum.

In RiskSpectrum PSA success in event trees can be included in the results. The method applied in RiskSpectrum has some limitations, and one of the most important simplifications is handled with the MCS BDD algorithm.

II. MCS AND BDD METHODS

In normal case Min Cut Upper Bound, MCUB, is a very good estimate of the top result. The condition is that the dependency between MCSs is reasonably low considering the probability of the events. In fact, if there are no dependencies between MCSs MCUB is exact. If the probabilities are low, the intersection between the MCSs also becomes insignificant (which is why MCUB and 1st order approximation are both normally good approximations).

However, if there is a significant amount of high probability events, and the MCSs that reside these high probability events are dependent – then MCUB can be overly conservative (not as much as 1st order approximation).

In RiskSpectrum PSA success events in event trees and mutually exclusive events are treated with simplified approaches. The treatment of success events in RiskSpectrum PSA is accomplished by generating MCSs from the coherent part of the fault tree (Ref. 5) and for each sequence add a success probability of not failing top events in the event tree (function events in RiskSpectrum terminology). The success probability is generated by analysing the success events under an OR gate, generating the failure events and calculating the probability as 1-P(failure). The success probability is stored in a separate event, referred to as a success module. This treatment is called simple success method. The simplifications are mainly that:

- the success event is quantified as a separate event, and not conditional on the events in the MCS
- that the success event is generated as a separate FT
- that the success event is quantified using MCUB

The simple success method gives an estimate of the success probability, but it is not exact. The difference compared to an exact treatment derives from the above simplifications. The simple quantitative method yields normally a good approximation for the current sequence, but due to the unconditional quantification of the success event it can yield an estimate that is above the default quantification method (Logical ET success), which does not take the success probability into account, when several sequences are calculated together.

Mutual exclusivity is calculated in the MCUB approach by creating partitions within the MCS list, where different mutual exclusive events are represented. This approach is conservative.

The calculations of the success treatment within event trees, as well as treatment of mutual exclusivity, are enhanced with the MCS BDD algorithm.

III. DESCRIPTION OF THE MCS BDD METHOD

III.A. General Description of the Method

The general process of the RiskSpectrum MCS BDD was described by (Ref. 6) and this paper is a continuation of that paper focusing on some details. First, however, a brief introduction to the method is presented.

The MCS BDD algorithm uses a pivotal decomposition method to construct the BDD structure from the MCS list. By continually picking up pivotal elements (decision variables) from the MCS list, both failure and success branches are being further developed. When all the branches have reached "1" (failure state) or "0" (success state), the BDD structure is generated completely. An important process of building the BDD structure is to select the pivotal element. The pivotal element is selected from the remaining MCS list and appended to the branch of the current pivotal node under construction. The order of pivotal elements is defined dynamically by an algorithm that is based on number of replications of the event and the cutset order.

Each node that is added is either included as an *Exact* node, or as an *Approximate* node. The concept of Exact and Approximate is described in more detail in section III.B. The process of defining what type of method (Exact or Approximate) is in the RiskSpectrum implementation equally important to the selection of pivotal element. The use of Approximate method is our way of ensuring a scalability of the MCS BDD method.

For each node included in the BDD, the MCS list has to be minimized. We use a simplified process for minimization of the MCS list, as a complete minimization in each step would be very time consuming.

All sequences using success modules are quantified separately and then the results are merged. MCSs generated from different initiators are treated separately.

III.B. Simplifications

III.B.1. MCS BDD or BDD from FT

The most obvious simplification in the method, compared to a complete analysis, is that we base the analysis on the MCS list. This means:

- That the complete cut set list is truncated, given that the MCSs are generated with a cut off
- That the rules for generation of MCSs are followed

The truncation issue is simply a matter of setting an appropriate threshold for the calculation. The use of truncation also means that sensitivity and uncertainty analysis results could potentially be affected (Risk Increase Factor and uncertainty results, when there are very wide distributions).

The generation of MCSs follows the rules of MCS generation. For RiskSpectrum the NOT logic is used to exclude unwanted combination of events. The MCSs are generated from the coherent part of the fault tree and then verified with the complete structure. If a complete solution of the fault tree would be applied, for example BDD, also the NOT logic would affect the probability of gates. If the fault tree is coherent, and a complete set of MCSs are generated, the result of the BDD and the MCS BDD would be the same.

A special case in RiskSpectrum of using NOT logic is the use of *Success Modules*. The success modules (representing success within event trees) are possible to include in the MCS list.

III.B.2. Simplifications within the MCS BDD Method

In the setup and quantification of the BDD structure some simplifications are used. The simplifications are used to allow for a rapid set up of the BDD structure but still with sufficient accuracy.

The simplifications used in the analysis are:

- Split of the MCS list into a part treated by BDD and a part treated by MCUB (as in the standard RiskSpectrum PSA quantification)
- Use of exact and approximative BDD nodes
- Not having a complete treatment of dependencies between events in the success module and failure part of MCS list (conservative)

The simplifications are possible to set by the user using three factors

- MCS threshold, given in percent of top, above which MCSs are treated by MCS BDD
- The application of exact and approximate BDD, as well as treatment of dependencies between events in success and failure part, are governed by a combination of Fussel Vesely importance and event probability.
- The precision of the success module calculation will affect the precision of the results. The impact of conservative estimates of the success module will be more significant when the accuracy in the top results is increased, and hence it may be relevant to adjust the precision of the success module quantification.

III.C. Exact, Approximative and ZBDD Node Treatment

The exact method is directly following from Shannon non-intersect decomposition,

$$f = xf_1 + \overline{x}f_0,$$

where x is one of decision variables. The functions f_1 and f_0 are Boolean functions evaluated at $x=1$ and $x=0$ respectively. The quantification of the node is performed as

$$P_f = P(X) \cdot P_{f1} + (1-P(X)) \cdot P_{f0}$$

Where P_{f1} and P_{f0} is the probability calculated for those nodes.

The approximate treatment means that when the MCS is analyzed and the BDD structure is built, then in the failed part of the structure (of the current pivotal element) only MCSs where the failed event is included are considered. This is following the same principle as ZBDDs. Exact nodes and approximate nodes can be mixed.

The quantification in the approximate method is performed as:

$$P(Top) = P(X) \cdot [\, P(MCS\ including\ X) + P(MCS\ not\ including\ X) - P(MCS\ including\ X) \cdot P(MCS\ not\ including\ X)\,] + (1 - P(X)) \cdot P(MCS\ not\ including\ X)$$

Compared with the ZBDD method subtraction of the intersection between P(MCS including X) and P(MCS not including X) would not be performed. The approximate solution is therefore yielding a more accurate result than ZBDD, but there are conditions that have to be applied to make the approximate method valid. The approximate method cannot be used for nodes when:

- An event is negated in the MCS structure
- When the success module contains event(s) that should be merged with the BDD structure, those nodes have to be represented before any approximate node is inserted.

The last bullet has been proved to be a significant drawback of the approximate method, as it adds a restriction to the dynamic allocation of BDD nodes.

The advantage of the use of the Approximative method in MCS BDD is exemplified by the results presented in TABLE I and TABLE II. TABLE I presents the characteristic of the MCS list (amount of MCS and MCUB top result) and also the primary events in the master fault tree used for generation of the MCS list. A primary event is any event in the fault tree used for calculation. A primary event might therefore be a module of several events.

TABLE I. General characteristics of the MCS lists

	Primary Events	MCS	MCUB top	Comment
Example 1	117	26421	5,1215E-04	Multiple IE
Example 2	3091	4798	1,1602E-05	Single IE
Example 3	13090	26429	1,6516E-06	Multiple IE
Example 4	872	9899	1,0006E-05	Single IE, high conditional probability

TABLE II. Results of and characteristic of the MCS BDD

	BDD Calculation setting			Items used in BDD				Results	
	MCS cutoff	FQ	FV	MCS	Used events	Exact treated events	Nodes	BDD setup time	Top result
Example 1	*MCUB*								*5,1215E-04*
	1,00E-03	1,00E-03	1,00E-04	325	64	58	826	8,00E-03	5,0047E-04
	1,00E-05	1,00E-03	1,00E-04	2799	116	58	1703	2,30E-02	4,9991E-04
	1,00E-06	1,00E-03	1,00E-04	4602	117	58	2393	3,60E-02	4,9991E-04
	1,00E-06	1,00E-06	1,00E-06	4602	117	115	3337	5,70E-03	4,9990E-04
	1,00E-10	0	0	17069	117	117	9598	4,80E-01	4,9990E-04
	0	0	0	26421	117	117	8720	1,26	4,9990E-04
Example 2	*MCUB*								*1,1602E-05*
	1,00E-02	1,00E-02	1,00E-02	1176	297	11	904	1,60E-02	1,0794E-05
	1,00E-03	1,00E-02	1,00E-02	4060	439	11	5415	9,00E-02	1,0782E-05
	1,00E-04	1,00E-02	1,00E-02	4716	462	11	6244	1,11E-01	1,0782E-05
	1,00E-04	1,00E-03	1,00E-03	4716	462	60	129175	4,911	1,0779E-05
	1,00E-04	1,00E-04	1,00E-04	4716	462	218	971741	88,7	1,0778E-05
Example 3	*MCUB*								*1,6516E-06*
	1,00E-02	1,00E-02	1,00E-02	24412	1535	516	28721	7,39E-01	1,2081E-06
	1,00E-03	1,00E-02	1,00E-02	26217	1566	516	30524	7,66E-01	1,1974E-06
	1,00E-04	1,00E-02	1,00E-02	26408	1567	516	30568	8,66E-01	1,1963E-06
	1,00E-04	1,00E-03	1,00E-03	26408	1567	1003	52137	1,293	1,1951E-06
	1,00E-04	1,00E-04	1,00E-04	26408	1567	1355	55642	1,376	1,1951E-06
	0,00E+00	1,00E-04	1,00E-04	26429	1567	1355	55781	1,381	1,1950E-06
	0,00E+00	0,00E+00	0,00E+00	26429	1567	1567	60864	1,45	1,1950E-06
Example 4	*MCUB*								*1,0006E-05*
	1,00E-02	1,00E-02	1,00E-02	189	104	26	186	9,00E-03	8,2627E-06
	1,00E-03	1,00E-02	1,00E-02	726	227	26	865	9,00E-03	8,1880E-06
	1,00E-04	1,00E-02	1,00E-02	2241	414	26	3401	3,90E-02	8,1803E-06
	1,00E-04	1,00E-03	1,00E-03	2241	414	66	11735	1,08E-01	8,1802E-06
	1,00E-04	1,00E-04	1,00E-04	2241	414	160	21736	2,01E-01	8,1802E-06
	0	1,00E-04	1,00E-04	9899	612	160	1965378	109,9	8,1794E-06

It can be noticed that the results of the MCS BDD (total result, including part treated with normal MCUB) should be converging quickly towards results with a high accuracy. All cases produce results that are reasonably close to the accurate results already using the first applied setting (1E-2, 1E-2, 1E-2). When a truncation for the BDD of 1E-3 is applied, then the results are more or less exactly representing the correct results.

The number of nodes in the BDD is referring to a combination of the number of MCSs, the total number of events included and the number of exactly treated nodes. Even though the examples above are from real PSA problems, different models will generate a unique problem. The scalability in the algorithm lies in the possibility to apply the Approximative method, which will enable a sufficient accuracy.

III.D. Quantification of Success Modules

The success module is calculated in RiskSpectrum PSA and a separate MCS list is provided for each success module. If the success module is independent of the failure MCS list, it can be treated as a separate event. The numerical quantification of the success module should however always be performed as accurate as possible, to avoid underestimate of the success probability.

When there are dependencies between the failure part of the MCS list and the success module, an increased accuracy in the calculation can be achieved by considering the dependency. This is achieved by merging the success module BDD with the failure MCS BDD. It can be noticed that the parts of the BDD structure where dependency is not considered relevant (no dependency or when the impact of the dependency is negligible), those events can be grouped together as a separate event.

Dependencies between the success module and the failure MCS is increasing the complexity and the expected size of the BDD. The reason is that during the build of the BDD structure (including merge of the success module) it has to be ensured that all nodes relevant in both structures (failure MCS and success module) are represented before any approximative BDD node is included. The condition for the approximate solution is that the *MCS including X* and the *MCS not including X* are coherent – if not, the intersection cannot be subtracted. If the success module contains events that may decrease the probability of a path it therefore violates the condition. If all relevant events from the success module are already included in the BDD, prior to use of approximate method – then the success module is already estimated. If the events in the success module are not included in the failure part of the BDD structure, then the success module could be considered as a separate failure event – independent of the rest of the BDD – and it then also satisfies the condition.

The restriction applied in the method is to ensure that all relevant events in the success module are considered first in the BDD structure. This significantly affects the efficiency of dynamic selection of pivotal element as many paths have to be considered. Tests have been performed comparing results with only exact BDD or using ZBDD instead. The exact BDD is preferable, but the quantification time may be significantly affected. The use of ZBDD instead of approximate method does not have the same drawback (as there is no subtraction) and therefore ZBDD could be used together with exact treatment. The tests performed have shown that the method would yield estimates that are higher than MCUB in the normal algorithm in several cases. Both these alternative solutions have therefore been disregarded as not practical.

To be able to use approximate method but to reduce the negative effect, a simplification has been introduced where only the events that have a significant impact on the global results are considered relevant for dependency. If an event has a low global impact, then a conservative treatment is acceptable. If an event is not included in the failure MCS, then dependency is not relevant to consider.

One method to estimate the impact of events on the total result including success modules, which has shown to be efficient in the MCS BDD, is to use a modified version of the Fussel Vesely measure. The modification to the normal FV estimate is to calculate the FV for each event in each success module. Each success module is also estimated with regard to its FV. The total FV for an event A is then estimated as:

$$FV_{totA} = FV_A + \Sigma\,(FV_{AinSMi} \cdot FV_{SMi})$$

Where FV_{totA} is the global estimate of FV for event A also including contents of success modules, FV_{AinSMi} is the FV for A in the quantification of top value for success module i and FV_{SMi} is the FV for the success module in the MCS list. It can be noticed that this is overestimating the importance of event A, as events included in the success module and in the normal structure are affecting the total results in different ways. However, the method is not underestimating the importance of any event. The overestimate of FV has to be considered if a ranking of events is performed, but the FV calculated in this method is intended to be compared with a threshold defining if the event should be treated exactly or not.

III.E. Selection of Pivotal Element Quantification Method

As was shown with the example in section III.C, the number of events for which an exact BDD structure is set up is affecting the time needed to build the BDD. Some events have to be treated exact, as presented in section III.C.

Assuming that the events do not have to be treated *Exact*, when is it acceptable to use the approximate method? Looking at the method for approximate, it can be noticed that:

- If the *MCS including X* is independent from *MCS not including X* the method is also exact.
- If there is a dependence (which can be expected in large PSA), there may be an overestimate of the results

The overestimate is dependent on the importance of the node, as an event with low impact on the total result will have small impact on the overestimate. The overestimate is also dependent on the probability of the event, as the overestimate relates to the intersection between *MCS including X* and *MCS not including X* multiplied with P(X). Based on these two observations, the FV and the probability of event X are candidates for selection of calculation method.

There are some possible methods for using these observations;

- Only use either of the measures (FV / Q)
- Use both of the measures and as soon as one indicates exact, then exact method is applied
- Combine the measures

In the implementation the two last options have been tested. The one that looks most promising is a combination of the measures: An event is treated with *Exact* treatment when it has a high probability and a high importance. In the approach, the user specifies a FV, for example 1E-2, and a probability, for example 1E-3. All events where a balanced measure of Q and FV is above 1E-5 (1E-2 · 1E-3) are treated exact. For example, an event with a probability of 1E-4 is treated exact if its FV is above 0.1.

It shall be noticed, as discussed in section III.D that the FV measure also has to cover the importance of events within the success modules.

III.E. Calculation of Mutual Exclusivity and Basic Event to Basic Event Relation

In RiskSpectrum PSA there are a number of new features that have been introduced lately, which are not standard in ET/FT tools. The basic event to basic event relation constitutes a method where the probability of one event is dependent on one or several other basic events. The mutual exclusivity feature allows a user to define MUX groups, ie groups in which events are to be considered mutual exclusive to each other in the calculation.

The feature basic event to basic event relation will not affect the MCS BDD feature, as the probability of each event is not dependent on the position in the BDD structure.

The feature mutual exclusivity requires a new framework. First, there are different types of mutual exclusivity in the MCS list. Each frequency event (initiating event) is considered to be mutual exclusive to any other frequency event. A cut set list for an individual sequence, is mutually exclusive versus another cut set list if the cut set list is also considering the success probability of function events. These two types of mutual exclusivity are easily handled within the MCS BDD approach. The MCS list is first subdivided into sub sets of MCS list, each representing each sequence or initiating event. If there is a combination of initiating events and sequences, each combination of initiating event and sequence is one sub set. For each sub set of MCS list, a BDD is built. It shall be noticed, that a sequence considering success is, in RiskSpectrum, represented by success modules. Each success module is a unique combination of success top events (function events) and thereby representing a unique sequence.

In contrast to the initiating event and sequences, mutual exclusivity between individual events cannot be considered in the same way. This is due to the dependencies between MCSs containing the MUX events.

The mathematical framework considered is based on following

Let x_i denote the events that are considered MUX versus each other. The MCS list is denoted by $\boldsymbol{C}$. A subset C' of $\boldsymbol{C}$ is representing all MCSs not containing any event x_i. The sub set C_i of $\boldsymbol{C}$ is containing all MCSs that are containing event x_i and C' (that is, containing the MCSs where other x_i is not included).

The calculation of the top probability, considering the mutual exclusivity for the MCS list would be:

$$P_{top} = \Sigma(P(x_i) \cdot P(C_i)) + (1-\Sigma P(x_i)) \cdot P(C')$$

$P(x_i) \cdot P(C_i)$ then represents the value of the MCS list given that all other events x_i are FALSE. If $\Sigma P(x_i) = 1$, then inclusion of the quantification $P(C_j)$ will already be represented by $\Sigma(P(x_i) \cdot P(C_i))$.

Inclusion of the framework in the BDD will require a new type of MUX node. The constraints are:

- That the events in a MUX group are calculated sequentially
- That a member of a MUX group can only be member of one MUX group
- That the MUX events are not negated (they can however be included in a success module)

IV. VERIFICATION OF THE METHOD

A new calculation algorithm requires significant testing of its stability and that the results generated are as expected. It shall be noticed that the calculation of event probability and modularisation of the fault tree is performed with the same parts as the normal RiskSpectrum calculation algorithm.

The verification procedure of the MCS BDD is

- describing the mathematics of the algorithm at conferences
- code review of the implementation
- to ensure a proper representation of the mathematics in the BDD (exact and approximate)
- to ensure that the implementation is correct with regard to treatment of MCS

A proper representation of the mathematics of the BDD for coherent fault trees is achieved by:

- verification of test cases that can be calculated by hand (exact and approximate)
- comparison with large scale models where the MCUB, 1st -3rd order results are close to expected
- comparison with large scale models where the MCUB, 1st -3rd order results are not close to expected

A proper representation of the mathematics of the BDD for event trees including success is achieved by:

- verification of test cases that can be calculated by hand (exact and approximate)
- comparison with large scales models, using the simple quantitative approach
- calculating all sequences in large event trees, summing the sequences. The result should be the same (very close) to the initiating event frequency
- comparison with another method to quantify the top result

In this paper we focus on the comparison with another method to quantify the top result, as the rest of the tests are reasonably self-explaining. The method selected for comparison is the method presented in (Ref. 7). The main advantage of the method is that it enables the use of the normal RiskSpectrum calculation algorithm for coherent fault trees, and thereby there will be no differences due to reliability models, approaches with regard to modularisation of master fault tree or how the master fault tree is set up.

The method presented in (Ref. 7) is based upon following observation:

$$P(Q \cdot -S) = P(Q) - P(Q \cdot S)$$

As exemplified in (Ref. 7) for a sequence of top events (function events) A, B and C (A fails, but B and C are in success):

$$P(A \cdot -B \cdot -C) = P(A) - P(A \cdot C) - P(A \cdot B) + P(A \cdot B \cdot C)$$

It shall be noticed, that the current implementation for definition of success modules is not completely exact (it applies cut off and the success module is considering as a FT using the inputs to the top events). This means that small deviations may be expected, but these deviations shall be explainable.

An example is used to illustrate how the comparison is used to verify the MCS BDD results with the method proposed in (Ref. 7). The function events in this example are taken from real scale PSA models, which include dependencies and sufficient complexity.

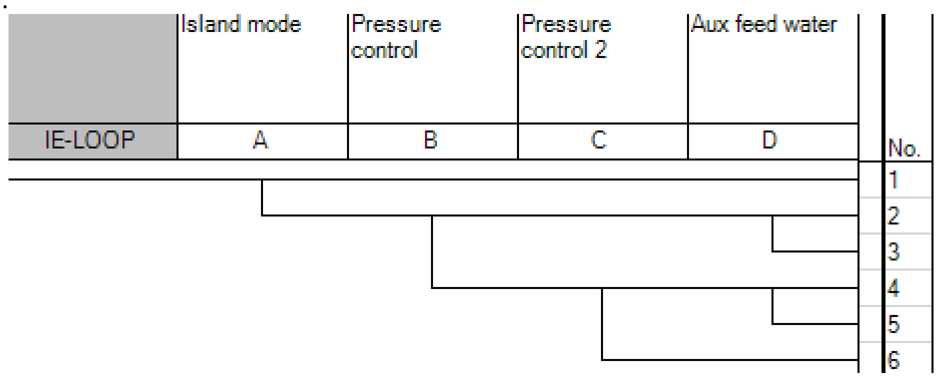

Fig. 1. Illustration of a small test case

Three sequences chosen for this example are; sequence 1, sequence 2 and sequence 5.

Sequence 1	$P(IE \cdot -A) = P(IE) - P(IE \cdot A)$
Sequence 2	$P(IE \cdot A \cdot -B \cdot -D) = P(IE \cdot A) - P(IE \cdot A \cdot D) - P(IE \cdot A \cdot B) + P(IE \cdot A \cdot B \cdot D)$
Sequence 5	$P(IE \cdot A \cdot B \cdot -C \cdot D) = P(IE \cdot A \cdot B \cdot D) - P(IE \cdot A \cdot B \cdot C \cdot D)$

The results using method in (Ref. 7) above, using standard logical and simple quantitative in RiskSpectrum and using MCS BDD are presented in TABLE III for the model illustrated above in Fig. 1.

TABLE III. Results of the example in Fig. 1.

	Logical&Simple	MCS BDD	Ref. 7
Seq 1	2,23E-01	2,22E-01	2,22E-01
Seq 2	7,81E-02	7,81E-02	7,80E-02
Seq 5	1,12E-08	1,09E-08	1,09E-08

In this small illustration of a comparison with an alternative method, the results using both the standard Logical and Simple Quantitative approach in RiskSpectrum as well as the MCS BDD are aligned with the results of the alternative method. It can be noticed that the MCS BDD is generating the same results (with the accuracy considered) as the alternative method for sequences 1 and 5. For sequence 2 there is a slight difference, which is explained by the accuracy used in the generation of the success module and not really relating to the MCS BDD algorithm. It may also be noticed that the difference between Logical and Simple Quantitative and the other methods are small for the studied examples.

V. CONCLUSIONS

The MCS BDD refines the calculation of the MCS list. The improvements lie mainly in an improved accuracy in the calculation of events with high probability (or high importance), treatment of dependencies between MCSs and also treatment of the dependency with the contents of the success module.

The paper presents the importance of the approximative method in the building of the BDD to make the approach scalable, and it also discusses the process to determine when to use exact and when to use approximative method. A framework for including mutual exclusivity is presented.

Finally, the approach for how the MCS BDD is verified is discussed. A comparison with an alternative method, which is used in the verification process of success path treatment, is illustrated.

REFERENCES

1. R. BRYANT, “Graph Based Algorithms for Boolean Function Manipulation,” *IEEE Transactions on Computers*, 35(8):677-691, August 1986.
2. A. RAUZY, “Binary Decision Diagrams for Reliability Studies,” *Handbook of Performability Engineering*, pp. 381 – 396, 2008.
3. O. COUDERT and J.-C. MADRE, “Fault Tree Analysis: 10^{20} Prime Implicants and Beyond,” *Annual Reliability and Maintainability Symposium*, 1993.
4. A. RAUZY, “New Algorithms for Fault Trees Analysis”, *Reliability Engineering & System Safety*, 59(2), pp. 203–211, 1993.
5. O. BÄCKSTRÖM and P. KRCAL, “A Treatment of Not logic in Fault Tree and Event Tree Analysis”, *PSAM 11*, 2012
6. O. BÄCKSTRÖM, P KRCAL and W. WANG, “Quantification of MCS with BDD, accuracy and inclusion of success in the calculation – the RiskSpectrum MCS BDD algorithm” *PSAM 12*, 2014
7. O. NUSBAUMER and A. RAUZY, “Fault Tree Linking versus Event Tree Linking Approaches: a Reasoned Comparison,” *Risk and Reliability*, 277(3), pp. 315 – 326, 2013.