Improved Accuracy in Quantitative Fault Tree Analysis

R M Sinnamon and J D Andrews
Department of Mathematical Sciences
Loughborough University of Technology
Loughborough, Leics LE11 3TU.

Summary

The fault tree diagram defines the causes of the system failure mode or "top event" in terms of the component failures and human errors, represented by basic events. By providing information which enables the basic events' probability to be calculated the fault tree can then be quantified to yield reliability parameters for the system.

Fault tree quantification enables the probability of the top event to be calculated and in addition its failure rate and expected number of occurrences. Importance measures, which signify the contribution each basic event makes to system failure can also be determined. Due to the large number of failure combinations (minimal cut sets) which generally result from a fault tree study, it is not possible using conventional techniques to calculate these parameters exactly and approximations are required. The approximations usually rely on the basic events having a small likelihood of occurrence. When this condition is not met it can result in large inaccuracies. These problems can be overcome by employing the Binary Decision Diagram approach. This method converts the fault tree diagram into a format which encodes Shannon's decomposition and allows the exact failure probability to be determined in a very efficient calculation procedure.

This paper describes how the Binary Decision Diagram method can be employed in fault tree quantification.

1. Introduction

The Binary Decision Diagram (BDD) method, developed by Rauzy (Ref. 1), converts the fault tree to a binary decision diagram which encodes an If-Then-Else (**ite**) structure. An attractive feature of the BDD method is that the **ite** structure derives from Shannon's formula (Ref 2), such that if f(**x**) is the Boolean function for the fault tree top event then by pivoting about any variable X1 the Shannon formula can be written as:

$$X1.f1+\overline{X1}.f2 \qquad (1)$$

where f1 and f2 are Boolean functions with X1=1 and X1=0 respectively which are of one order less than f. The corresponding **ite** structure is **ite**(X1, f1, f2). A detailed account of this procedure is given in Ref 3 and Ref 4.

The size of the resulting BDD is very dependent upon the ordering chosen for the basic event pivot variables in the fault tree. Usually a Top-Down ordering is used, where the basic events occurring higher up the tree are considered as being less than those which occur lower down. This ordering of variables is discussed in Ref. 3 and covered in more detail in the paper, "Improved Efficiency in Qualitative Fault Tree Analysis" (Ref 5).

Example

Consider the simple fault tree illustrated in figure 1.

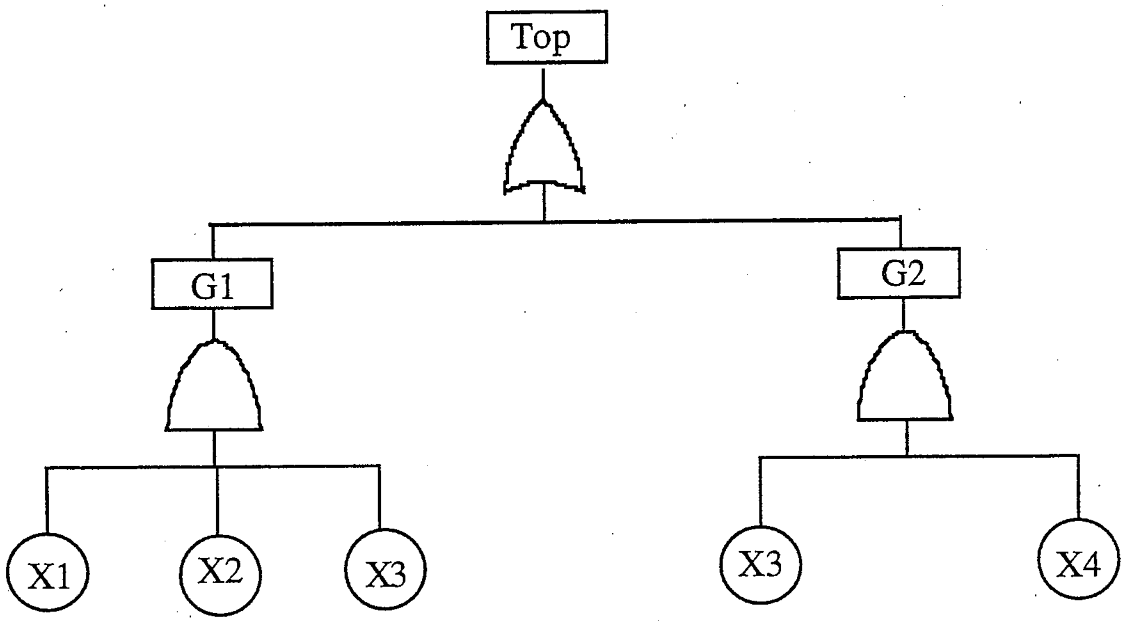

Figure 1. Example Fault Tree

The minimal cut sets for this fault tree are:

(1) {X1, X2, X3}
(2) {X3, X4}

Therefore, its structure function, $\phi(\mathbf{x})$ is :

$$\phi(\mathbf{x})=1-(1-X1.X2.X3)(1-X3.X4) \qquad (2)$$

A Top-Down ordering of basic events would give;

$$X1 < X2 < X3 < X4 \qquad (3)$$

since all basic events are on the same level. This ordering will yield the BDD shown in figure 2 (figure 2a with the Boolean equations to show its development from the structure function and the simplified form in figure 2b).

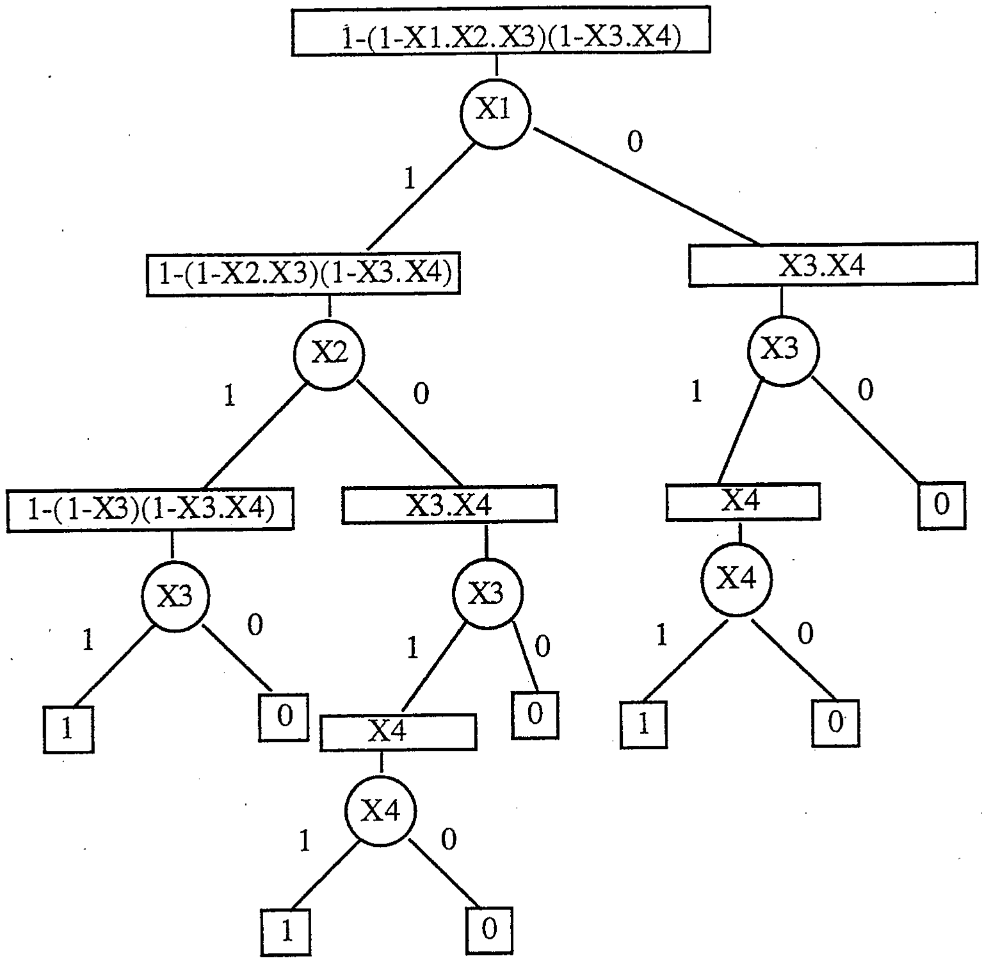

Figure 2a. BDD with Boolean Equations

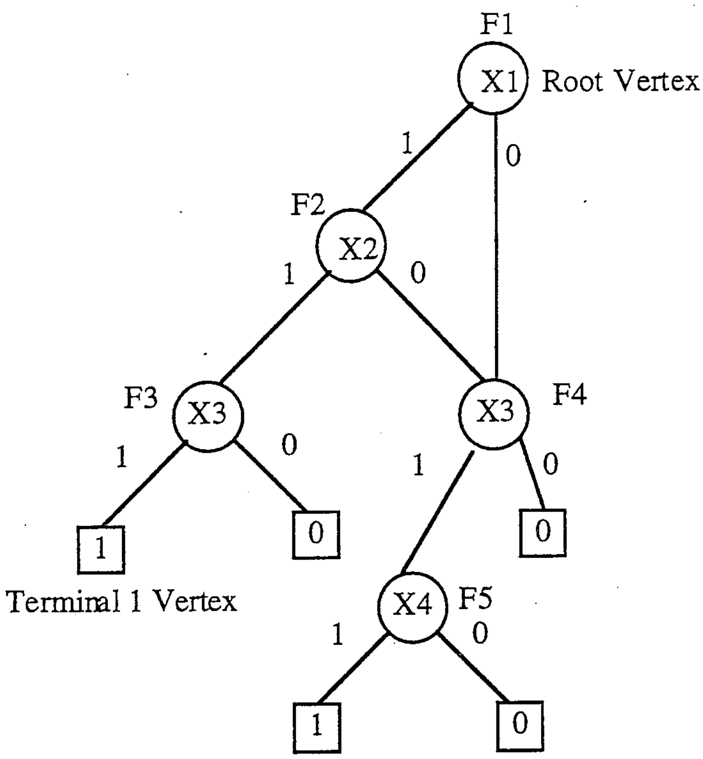

Figure 2b. BDD for the fault tree shown in figure 1

To obtain the cut sets of the fault tree paths are traced through the BDD from the top or root vertex to a terminal 1 vertex. Only the basic events that lie on a 1 branch (indicating the failure of that basic event) for each component are included in the cut set. Therefore the cut sets of the BDD shown in figure 2b are:

(1) X1.X2.X3
(2) X1.X3.X4
(3) X3.X4

The resulting BDD for this basic event ordering is not minimum, as it produces one redundant cut set. To obtain the minimal cut sets either the resulting cut sets can be reduced or the BDD can undergo a minimising procedure, whose details are given in Ref. 3. However for the purposes of quantification, the non-minimal BDD is used.

2. Top Event Probability

The conventional approach (see Andrews and Moss Ref (6), Henley and Kumamoto Ref (7)) to obtain the exact probability of the top event is to use the formula:

$$P(Top) = \sum_{i=1}^{nc} P(C_i) - \sum_{i=2}^{nc} \sum_{j=1}^{i-1} P(C_i \cap C_j) + \ldots\ldots + \ldots\ldots (-1)^{nc-1} P(C_1 \cap C_2 \cap \ldots\ldots \cap C_{nc}) \qquad (4)$$

Where C_i, i=1,..........nc are the minimal cut sets of the top event.

Clearly if the fault tree has many minimal cut sets, calculating P(Top) will require extensive calculations to evaluate each term in expression (4). For many complex fault trees this processing requirement is beyond the capability of the available machines. In order to reduce the calculations to a practical size approximations can be used. The Rare Event Approximation, $P_{RE}(Top)$, is commonly used:

$$P_{RE}(Top) = \sum_{i=1}^{nc} P(C_i) \qquad (5)$$

However a more accurate approximation is the Minimal Cut Set Upper Bound:

$$P_{MCSUB}(Top) = 1 - \prod_{i=1}^{nc} (1 - P(C_i)) \qquad (6)$$

The Binary Decision Diagram method, avoids the need to utilise these approximations as the probability of the top event is obtained directly from the diagram.

Since the **ite** structure for the top event of a fault tree is derived from Shannons' formula, i.e., if f(**x**)=**ite**(X1, f1, f2) then the corresponding Boolean function or structure function is f(x)=$X1.f1 + \overline{X1}.f2$. When a Boolean function is expressed in this form the probability of the top event is obtained by taking the expectation of each term:

$$E[f(\mathbf{x})] = q_1.E[f1] + (1 - q_1).E[f2] \qquad (7)$$

where q_1=E[X1], the probability that event 1 has occurred.

Therefore the probability of occurrence of the top event (Q_{sys}), can be expressed as the sum of the probabilities of the disjoint paths through the BDD. The disjoint paths through the BDD are found by simply including in a path from the root vertex to a terminal 1 node, all events which lie on the 1 branch and the 0 branches for each the basic events. Basic events which lie on a 0 branch are indicating in the paths as $\overline{Xi}$. Disjoint paths through the BDD shown in figure 2b are:

(1) X1.X2.X3
(2) X1.$\overline{X2}$.X3.X4
(3) $\overline{X1}$.X3.X4

Before performing with the calculation of Q_{sys} the basic events in the fault tree need to be assigned probabilities. For the fault tree shown in Figure 1 the component data provided on table 1 will be used.

basic event i	q_i	λ_i	$w_i = \lambda_i(1-q_i)$
X1	0.004	1.2E-6	4.8E-9
X2	0.003	4.3E-6	1.29E-8
X3	0.002	2.5E-6	5.0E-9
X4	0.001	3.1E-6	3.1E-9

Table 1. Basic Event Data.

In table 1 $\mathbf{q_i}$ is the Unavailability of component i, λ_i is the Conditional failure intensity of component i, and $\mathbf{w_i}$ is the Unconditional failure intensity of component i.

Using this data the top even probability Q_{sys} can be calculated as follows;

$$\begin{aligned} Q_{sys} &= P(X1.X2.X3 + X1.\overline{X2}.X3.X4 + \overline{X1}.X3.X4) \\ &= q_{X1} \cdot q_{X2} \cdot q_{X3} + q_{X1} \cdot (1-q_{X2}).q_{X3} \cdot q_{X4} + (1-q_{X1}).q_{X3} \cdot q_{X4} \\ &= 2.023976\text{E-}6 \end{aligned}$$

3. Unconditional System Failure Intensity

For some systems it is required to calculate the unreliability for the top event i.e., the probability it will not work **continuously** over a given time period. An upper bound for this is the expected number of top event occurrences, W(0, t):

$$W(0,t) = \int_0^t w_{sys} dt \tag{8}$$

where w_{sys} is the system unconditional failure intensity. This can be expressed as:

$$w_{sys} = \sum_i G_i(\mathbf{q}).w_i \tag{9}$$

where $G_i(\mathbf{q})$ is the criticality function for each component.

The criticality function $G_i(\mathbf{q})$ is defined as the probability that the system is in a critical state with respect to component i and that the failure of component i will then cause the system to go from the working state to the failed state, i.e., the probability that the system fails only if component i fails. Therefore:

$$G_i(\mathbf{q}) = Q(1_i, \mathbf{q}) - Q(0_i, \mathbf{q}) \tag{10}$$

Where:
$Q(1_i, \mathbf{q})$ – is the probability of system failure with q_i=1.
$Q(0_i, \mathbf{q})$ – is the probability of system failure with q_i=0.

Since Q_{sys} is a linear function in each q_i then $G_i(\mathbf{q})$, for each basic event can also be given by:

$$G_i(\mathbf{q}) = \frac{\partial Q_{sys}}{\partial q_i} \tag{11}$$

Evaluating each of the two terms $Q(1_i, \mathbf{q})$ and $Q(0_i, \mathbf{q})$ for each component could be achieved by first substituting q_i=1 and then q_i=0, i.e., the probability that component i fails is set to 1 and 0 respectively, and re-running all the system failure probability calculations. This would require the equivalent of 2n evaluations of the top event probability where n is the number of components in the system, to deduce all terms required in the expression for w_{sys} in eq. (9).

However a more efficient way of calculating the criticality function can be achieved directly from the BDD using the formula:

$$G_i(\mathbf{q}) = \sum_{xi} pr_{xi}(\mathbf{q})[po^1_{xi}(\mathbf{q}) - po^0_{xi}(\mathbf{q})] \tag{12}$$

Where:

$pr_{xi}(\mathbf{q})$ - is the probability of the path section from the root node to node xi (**Probprev**).
$po^1_{xi}(\mathbf{q})$ – is the probability of the path section from the 1 branch of node xi to a terminal 1 node (**Probpost 1 branch**).
$po^0_{xi}(\mathbf{q})$ – is the probability of the path section from the 0 branch of node xi to a terminal 1 node (**Probpost 0 branch**).
n – is the number of nodes in the BDD.

Using equation (12) calculating $G_i(\mathbf{q})$ requires one pass of the BDD to calculate $pr_{xi}(\mathbf{q})$, $po^1_{xi}(\mathbf{q})$ and $po^0_{xi}(\mathbf{q})$ for each node. With this information each $G_i(\mathbf{q})$ can be evaluated from eq. (12) and w_{sys} formed.

The algorithm Probpost used to calculate $po^1_{xi}(\mathbf{q})$ and $po^0_{xi}(\mathbf{q})$ is given in figure 3. The calculation of $pr_{xi}(\mathbf{q})$ can be achieved by the algorithm Probprev given in figure 4 and the criticality function $G_i(\mathbf{q})$ for each basic event is calculated as shown in figure 5.

Probpost(F)≡

Do for all F, end vertices to Root Vertex

F=ite(x, G, H)

$po^1_x(q)$*=prob(G)*

$po^0_x(q)$*=prob(H)*

insert in Probtable, R←Probtable(x, $po^1_x(q)$, $po^0_x(q)$*)*

Q←p(x).prob(G)+(1-p(x)).prob(H)

insert - in - computation - table ({<prob, F, ->, Q})

return R

return Q

next F

Figure 3. Probpost Algorithm

Set Probprev(Fi)=0 for all i.

Probprev(F)≡

start at Root Vertex, F

Probprev(F)=1

Add Probprev(F) to Probtable, i.e., Probtable(x, $po^1_x(q)$, $po^0_x(q)$, $pr_x(q)$*)*

Do for all F, Root Vertex to end vertices

F=ite(x, H1, H2)

if H1=0 or 1 Goto [A]

Probprev(H1)=Probprev(H1)+p(x).Probprev(F)

Add Probprev(H1) to Probtable

[A] if H2=0 or 1 next F

Probprev(H2)=Probprev(H2)+(1-p(x)).Probprev(F)

Add Probprev(H2) to Probtable

next F.

Figure 4. Probprev Algorithm

Set G(xi)=0 for all i
Do for all F
if F=Probtable(x, q1, q2, q3)
G(x)=G(x)+q3(q1-q2)
insert - in Criticality table G(x)
next F.

Figure 5. Algorithm for Calculating the Criticality Function G_{Xi}

The implementation of these algorithms can be demonstrated by their application to the example BDD given in figure 2. The **ite** table indicating how the BDD in figure 2b is stored within the computer program which performs these calculations is given in table 2:

Node Label	Variable	1 branch pointer	0 branch pointer
F1	X1	F2	F4
F2	X2	F3	F4
F3	X3	1	0
F4	X3	F5	0
F5	X4	1	0

Table 2. ITE table for the BDD

Performing one pass of the BDD to evaluate $po^1_{xi}(\mathbf{q})$ and $po^0_{xi}(\mathbf{q})$ for each node using Probpost gives:

Probpost(F5)
F5=**ite**(X4, 1, 0)
R←Probtable(X4, 1, 0)
Q←p(X4).p(1)+(1-p(X4)).p(0)=0.001

Probpost(F4)
F4=**ite**(X3, F5, 0)
R←Probtable(X3, prob(F5), 0)←(X3, 0.001, 0)
Q←p(X3).(0.001)+(1-p(X3)).(0)=2.0E-6

Probpost(F3)
F3=**ite**(X3, 1, 0)
R←Probtable(X3, 1, 0)
Q←p(X3).p(1)+(1-p(X3)).p(0)=0.002

Probpost(F2)

F2=**ite**(X2, F3, F4)

R←Probtable(X2, prob(F3), prob(F4))←(X2, 0.002, 2.0E-6)

Q=p(X2).(0.002)+(1-p(X2)).(2.0E-6)=7.994E-6

Probpost(F1)

F1=**ite**(X1, F2, F4)

R←Probtable(X1, prob(F2), prob(F4))←(X1, 7.994E-6, 2.0E-6)

Q=p(X1).(7.994E-6)+(1-p(X1)).(2.0E-6)=2.023976E-6

As can be seen the probability of the top event, Q (calculated for F1 above) agrees with the probability calculated previously using the disjoint paths of the BDD.

The values of Probpost 1 branch and Probpost 0 branch for each node are entered into the node probability table, PROBTABLE shown in figure 6.

Next Probprev is calculated and entered into the 5th column of the PROBTABLE.

Probprev Algorithm:

Probprev(F1)=Probprev(F2)=Probprev(F3)=Probprev(F4)=Probprev(F5)=0

Probprev(F1)=1

F1=**ite**(X1, F2, F4)

Probprev(F2)=0+p(X1).Probprev(F1)

=(0.004).(1)=0.004

Probprev(F4)=0+(1-p(X1)).Probprev(F1)

=(1-0.004).(1)=0.996

F2=**ite**(X2, F3, F4)

Probprev(F3)=0+p(X2).Probprev(F2)

=(0.003).(0.004)=1.2E-5

Probprev(F4)=0.996+(1-p(X2)).Probprev(F2)

=0.996+(1-0.003).(0.004)=0.999988

F3=**ite**(X3, 1, 0)

H1=1

H2=0

F4=**ite**(X3, F5, 0)

Probprev(F5)=0+p(X3).Probprev(F4)

=(0.002).(0.999988)=1.999976E-3

H2=0

F5=**ite**(X4, 1, 0)

H1=1

H2=0

PROBTABLE

Node Label	**Variable**	**post '1'**	**post '0'**	**Probprev**
F1	X1	7.994E-6	2.0E-6	1
F2	X2	0.002	2.0E-6	0.004
F3	X3	1	0	1.2E-5
F4	X3	0.001	0	0.999988
F5	X4	1	0	1.999976E-3

Probtable(i, 1)=Node Label
Probtable(i, 2)=Basic event of node Fi
Probtable(i, 3)=Probability of post '1' branch
Probtable(i, 4)=Probability of post '0' branch
Probtable(i, 5)=Probability of previous

Figure 6. PROBTABLE Array

Calculation of the criticality function is then straight forward using the algorithm provided in figure 5. The following values of the criticality function for each basic event are obtained.

G_{X1}=5.994E-6, G_{X2}=7.992E-6, G_{X3}=1.011988E-3, G_{X4}=1.999976E-3

Since we have calculated the criticality function for each component, the system parameter w_{sys} can now be evaluated using the basic event frequency data from table 1 and eq. (9).

$$w_{sys} = G_{X1}w_{X1} + G_{X2}w_{X2} + G_{X3}w_{X3} + G_{X4}w_{X4}$$

=1.1391734E-11

The expected number of top event occurrences in time, t, can be obtained from eq. (8).

4. Importance Measures

A very useful piece of information which can be derived from a fault tree study is the **importance** measure for each component or each minimal cut set. An importance analysis provides sensitivity measures which identifies weak areas of the system. This information can be very valuable particularly at the design stage. For each component its importance signifies the role that it plays in either causing or contributing to the occurrence of the top event. In general a numerical value is assigned to each basic event or minimal cut set which allows it to be ranked along with other failure events according to the extent of its contribution to the occurrence of the top event.

Probabilistic importance measures can be categorised in two ways: (i) those which are appropriate for system availability assessment (top event probability) and (ii) those which are concerned with system reliability assessment (expected number of top event occurrences), here we are concerned only with component measures in the first group.

(i) Birnbaum Measure of Component Importance

The Birnbaum Measure of importance (I_b) was first introduced in 1969 (Ref. 8). This measure is defined as the rate at which the system failure probability changes as the failure probability of component i changes:

$$I_{bi} = \frac{\partial Q_{sys}}{\partial q_i} \qquad (13)$$

As defined in eq. (11) I_{bi} is also the criticality function for component i, $G_i(q)$. The calculation procedure to evaluate the criticality function from the BDD structure has been described previously. Hence the calculation of this importance measure for each component is performed in the calculation of w_{sys} and can therefore be extracted from those calculations.

(ii) Criticality Measure of Component Importance

The criticality measure of importance (I_c) is defined as:

"The probability that the system is in a state at time t which is critical for component i **and** that component i has failed at time t conditional on system failure at time t."

$$I_{ci} = \frac{\{Q(1_i, \mathbf{q}) - Q(0_i, \mathbf{q})\} q_i}{Q(\mathbf{q})} = \frac{G_i(\mathbf{q}).\, q_i}{Q_{sys}} \tag{14}$$

All terms in eq. (14) have been evaluated in calculating Q_{sys} and w_{sys} and are readily available to determine this importance measure.

(iii) Fussell-Vesely Measure of Component Importance

This measure of Importance is usually close in numerical value to the criticality measure. The Fussell-Vesely Importance (I_{FV}) is defined as the probability of the Union of the Minimal Cut Sets which contain event i divided by the top event occurrence probability. I_{FV} therefore gives the probability that when the system fails, component i contributed to the failure. Calculating I_{FV} for each component requires the use of both the minimal BDD to track the minimal cut sets and the non-minimal BDD for the probability calculations. The calculation of I_{FV} is performed in 4 steps:

(1) First a counter for the number of times each variable is encountered in a minimal cut set is initialised.

(2) A search is then performed on **each** path of the **minimum** BDD which will correspond to a minimal cut set. For each variable in the path i.e. the path passes out of the 1 branch, the occurrence counter is incremented. Once the search has been completed, it is known how many minimal cut sets contain each basic event.

(3) For events with **more** than one occurrence a search is made for the nodes in the **non-minimal** BDD that have this variable. For each basic event:

$$I_{FVi} = \frac{\sum_{xi\,nodes} q_i\, pr_{xi}(\mathbf{q})\, po^1_{xi}(\mathbf{q})}{Q_{sys}} \tag{15}$$

(4) If a variable has just one occurrence in a minimal cut set, c_i then:

$$I_{FVi} = \frac{p(c_i)}{Q_{sys}} \tag{16}$$

5. Applications

The BDD quantification method was benchmarked against a test example fault tree called 'Dresden-3' used by Platz and Olsen (Ref. 9). The structure file and data file for this tree are contained in appendix A. The input file gives the gate name (a number over 1000), the gate type, the number of gate inputs, the number of basic event inputs, followed by a list of the inputs themselves. Each basic event in the fault tree has two lines of data to specify its failure and repair characteristics, the first line gives the name of the basic event and the model type, in this case model type F was used for each component. The second line gives, for this failure model, parameters, λ and τ (λ must be multiplied by 1E-6). λ represents the component constant failure rate and τ its mean time to repair (MTTR)

The following calculations are then used for the model type F, to obtain q_i and w_i for each basic event.

$$q_i=\lambda.\tau/(\lambda.\tau+1) \quad (17)$$

$$w_i=\lambda(1-q_i) \quad (18)$$

A summary of the quantification results is given in table 3. The code runs on a Sun workstation, the execution time is given in seconds.

Name	Dresden-3
No. of Gates	60
No. of Basic Events	57
No. of Minimal Cut Sets	11,934
Time (s)	0.6
Q_{sys}	4.70085E-7
w_{sys}	2.87887E-8

Table 3. BDD quantification results for Dresden-3 fault tree

As a comparison , Dresden-3 was analysed using a state-of-the art conventional Fault Tree Analysis package, whose results can be seen in table 4.

No. of Minimal Cut Sets	11,934
Time	4hrs 10min 28s
Q_{sys}	4.81119E-7
w_{sys}	2.97304E-8

Table 4. FTA quantification of Dresden-3 fault tree

Hence, for this example the BDD method is significantly faster than conventional quantification techniques. Also, along with great savings in computation time the BDD technique gives exact probability values for Q_{sys} and w_{sys}, whereas the conventional method results in a loss in accuracy of 2.34% and 3.27% respectively for these parameters.

6. Accuracy - Comparison with FTA

To compare the accuracy of the BDD technique with the conventional Kinetic Tree theory approach, 10 example fault trees were analysed, the results of which are given in table 5. Some of these benchmark fault trees are taken from industry and the others are produced as simple structures to test different aspects of the analysis code.

Tree	No. of Gates	No. of basic events	No. of Minimal Cut Sets	BDD Q_{sys}	FTA Q_{sys}	BDD w_{sys}	FTA w_{sys}
1	17	11	43	2.08587E-2	0.0209883	7.52376E-6	8.03221E-6
2	63	32	8,716	4.272258E-7	4.27248E-7	2.777932E-4	0.000277835
3	21	40	416	1.317774E-6	1.31778E-6	8.990849E-4	0.0008991
4	10	10	13	6.43795E-2	0.068559	1.96564E-4	0.000211151
5	4	6	3	3.39397E-8	3.4E-8	2.74191E-10	2.7474E-10
6	4	6	6	7.06927E-5	7.10911E-5	3.09498E-6	3.12839E-6
7	3	4	2	3.0776E-4	0.000308	6.07692E-7	6.0836E-7
8	10	8	10	1.23233E-5	1.2371E-5	2.27332E-7	2.28347E-7
9	3	4	2	2.02398E-6	2.02656E-6	1.13917E-11	1.1392E-11
10	30	60	7,056	4.371846E-7	4.17233E-7	3.092103E-4	0.000297212

Table 5. Quantification Results of 10 example fault trees

It is evident from the results in table 4 that when the fault tree has a small number of minimal cut sets the conventional approximation method has only a small error – an average over estimate of 1.22% for the system failure probability and 2.3% for the system unconditional failure intensity. However, for tree 10, which has a large number of minimal cut sets, there is an error of 4.56% for Q_{sys} and 3.88% for w_{sys}. These inaccuracies should be avoided when a risk assessment is performed on an industrial system.

7. Conclusion

The use of BDDs to improve the efficiency of calculating the minimal cut sets and prime implicants of a fault tree has been proven for large complex fault trees by Rauzy Ref. (1) and Ref. (4).

This paper extends the use of the BDD method to calculate top event parameters, such as system failure probability, failure intensity and expected number of top event occurrences. The added advantage of obtaining these parameters directly from the BDD, when compared to traditional Kinetic tree theory approach (Ref. (10) and Ref. (11)), is that the resulting values are exact. Approximations used in conventional fault tree analysis are shown to be inadequate for some fault trees.

The paper has also shown that the commonly used component Importance measures for top event probability can be calculated directly from the BDD. Further, the BDD method has proven to be extremely efficient as a means of quantification. Only one pass of the BDD structure is required to calculate all parameters.

8. References

1. A. Rauzy, "New algorithms for fault tree analysis," *Reliability Engineering and System Safety,* vol 40, 1993, pp203-211.

2. W.G. Schneeweiss, "*Boolean Functions with Engineering Applications and Computer Programs,*" Springer-Verlag, 1989.

3. R.M. Sinnamon and J.D. Andrews, "New Approaches to Evaluating Fault Trees," *Proceedings of Esrel'95 Conference,* June, 1995 pp241-254.

4. A. Rauzy et al., 'Computation of prime implicants of a fault tree within Aralia," *Proceedings of Esrel'95 Conference,* June, 1995 pp190-202.

5. R.M. Sinnamon and J.D. Andrews, "Improved Efficiency in Qualitative Fault Tree Analysis", *12th Arts, Advances in Reliability Technology Symposium,* Manchester, April 1996.

6. J.D. Andrews and T.R. Moss, *"Reliability and Risk Assessment,"* Longman Scientific and Technical, 1993.

7. E.J. Henley and H. Kumamoto, *"Reliability Engineering and Risk Assessment,"* Englewood Cliffs, 1981.

8. Z.W. Birnbaum, "On the Importance of Different Components in a Multicomponent System", in *Multivariate Analysis II, P. R. Krishnaiah (Ed),* Academic Press, 1969.

9. O. Platz and J.V. Olsen, *"FAUNET: A Program Package for Evaluation of Fault Trees and Network,"* Research Establishment, Risø Report No.348, DK-4000 Roskilde, Denmark, Sept. 1976.

10. J.B. Fussell and W. E. Vesely, "A new methodology for obtaining cut sets for fault trees", *Trans Am Nucl Soc,* Vol 15, 1972, June, pp262-263

11. W.E. Vesely, "A Time Dependent Methodology for Fault Tree Evaluation", *Nuclear Eng and Des,* 13, 1970, pp337-360.

Appendix A Input file for Dresden-3

1060	and	2	01058	1059	
1058	and	2	01057	1055	
1059	and	2	01053	1045	
1057	and	3	01037	1034	1032
1055	and	3	01026	1022	1019
1053	or	2	01052	1051	
1045	or	2	01044	1043	
1044	or	0	249	48	
1043	or	1	11042	47	
1052	or	0	257	56	
1051	or	1	11050	55	
1042	or	2	01041	1040	
1050	or	2	01049	1048	
1049	or	0	254	53	
1048	or	1	11047	52	
1041	or	0	246	45	
1040	or	1	11039	44	
1047	or	2	01057	1046	
1039	or	2	01055	1038	
1046	or	0	250	51	
1038	or	0	243	42	
1037	or	2	01036	1025	
1034	or	2	01033	1020	
1032	or	2	01056	1031	
1036	or	0	241	40	
1025	or	2	11024	1035	29
1033	or	0	237	36	
1020	or	0	224	23	
1056	and	2	01030	1028	
1031	or	0	235	34	
1024	or	0	228	27	
1035	or	0	239	38	
1030	or	2	01029	1009	
1028	or	2	01027	1017	
1029	or	0	233	32	
1009	or	1	11008	10	
1027	or	0	231	30	
1017	or	1	11015	18	
1015	or	1	11014	16	
1008	or	2	01007	1006	
1014	or	2	01013	1012	
1007	or	0	29	8	
1006	or	0	27	2	
1013	or	0	215	14	
1012	or	1	11005	13	
1005	or	2	01004	1003	
1004	or	0	26	5	
1003	or	2	01002	1001	
1002	or	0	24	3	
1001	or	0	22	1	
1026	or	2	01025	1023	
1022	or	2	01021	1020	
1019	or	2	01054	1018	
1023	or	0	226	25	
1021	or	0	222	21	
1054	and	2	01016	1011	
1018	or	0	220	19	
1016	or	1	11015	17	
1011	or	1	11010	12	
1010	or	1	11009	11	

Data file for Dresden-3

```
1         F
1,10
2         F
0.1,15
3         F
9.009,48
4         F
9.009,48
5         F
100,334
6         F
1,5
7         F
1,10
8         F
9.009,48
9         F
9.009,48
10        F
1,5
11        F
1,5
12        F
5,8
13        F
1,5
14        F
9.009,48
15        F
9.009,48
16        F
1,5
17        F
5,8
18        F
1,5
19        F
5,8
20        F
5,3
21        F
5,8
22        F
10,3
23        F
10,200
24        F
1,5
25        F
10,3
26        F
5,8
27        F
9.009,48
28        F
9.009,48
29        F
1,5
30        F
5,8
31        F
10,3
32        F
5,8
33        F
10,3
34        F
5,8
35        F
5,3
36        F
5,8
37        F
10,3
38        F
1,70
39        F
10,10
40        F
5,8
41        F
10,3
42        F
5,8
43        F
10,3
44        F
1,5
45        F
9.009,48
46        F
9.009,48
47        F
1,5
48        F
5,8
49        F
10,3
50        F
5,8
51        F
10,3
52        F
1,5
53        F
9.009,48
54        F
9.009,48
55        F
1,5
56        F
5,8
57        F
10,3
```