The Importance Analysis of Componments in the Aviation System Using the Binary Decision Diagrams

XU Hengcheng CHEN Pu
Dept. of Aerial Instrument and Electric Engineering
The First Aeronautical Institute of Air Force
Xinyang, China
xuhengcheng@sohu.com

ZHANG Jianguo
Dept. of Reliability Engineering Laboratory
Bei Hang University
Beijing, China

***Abstract*—as the aviation system becomes more and more complex, the conventional fault tree analysis (e. g, top-down and bottom-up) techniques, which used to calculate the component importance in the aviation system , is often faced the limitations in terms of the accuracy and efficiency. The Binary Decision Diagram (BDD) method, that can be implemented in the computer efficiently, has the characteristic of the high efficiency calculation and precise result. This paper will talk about the new method or analysis way (BDD) to calculate the component importance in the aviation system instead of the conventional fault tree analysis (FTA) method, in order to improve the aviation system's design and increase the aviation component's reliability.**

***Keywords*—BDD, Fault Tree, Structural Importance, Probabilistic Importance**

I. INTRODUCTION

The component importance, that describes the importance of a component or sub-system to the whole system in quantitative, is defined as the probability that a component or sub-system when failed causes the whole system failed, or the probability that a basic event when occurred causes the top event occurring in the fault tree. It is proved that the whole system hasn't the same sensitivity to every component or sub-system; some component when failed will cause the whole system failing immediately, some not, the component importance is a key factor to improve the whole system's design. The component importance measure, which gives an order of the component following by their importance, can provide a good advice to improve the whole system design, increase the whole system's reliability and settle a good test points with less manpower and material resources[1].

Conventional fault tree analysis (e.g. top-down and bottom-up) techniques are now commonly used to calculate the importance of the component in reliability terms. Because this way is based on the minimal cut sets, the determination of these minimal cut sets is a very time consuming procedure even on modern high-speed digital computer, this limitation is prominent especially when analyzing the component importance in the complex aviation system.

To overcome this limitation, a completely new approach, called the Binary Decision Diagram (BDD) method, has been introduced. BDD is a graph description of the Boolean Function, which is widely used in the network, electro-circuit, logic-integration and logic validate. Rauzy [1] first used the BDD method to analysis the fault tree, in reliability terms. Because the BDD method can analysis the fault tree more efficiently, accurately, and can be efficiently implemented in the computer [2-3], and now, it begins to be used in the large system's reliability analysis[2], [3].

This paper will describe this new method which called BDD to calculate the component importance in the aviation industrial system, and provides a new way to analyze the importance measure of components, in order to improve the aviation component's design and increase the aviation system's reliability.

II. IMPORTANCE MEASURES

The importance measure of the component can be categorized in two ways: (1) structural importance, and (2) probabilistic importance[4].

A. *Structural Importance*

Structural importance is used to assess the importance of a component's location to the system operation, without considering the probability of the failure of the component. It can be denoted as follow formula [5],[6]:

$$I_i^{\Phi} = \frac{1}{2^{n-1}} \sum_{2^{n-1}} [\Phi(1_i q) - \Phi(0_i q)] \tag{1}$$

Where :

I_i^{Φ} : The structural importance of the component i

n : Total number of components in system.

$\Phi(1_i, q) = (q_1, q_2, ..., q_{i-1}, 1, q_{i+1}, ..., q_n)$: The system un-reliability when it is known that component i is failed, i.e. $q_i = 1$.

$\Phi(0_i, q) = (q_1, q_2, ..., q_{i-1}, 0, q_{i+1}, ..., q_n)$: The system un-reliability when it is known that component i is operational, i.e. $q_i = 0$.

q_i : The state of component i, with only two state: $q_i = 1$ (failed), $q_i = 0$ (operational), $i = 1,2,...,n$.

Lambert [5] introduced an alternative way to calculate the structural importance. He let $q_j = \begin{cases} 1, j = i \\ 1/2, j \neq i \end{cases}$.

The structural importance of the component i can be described as:

$$I_i^{\Phi} = G_i(q) = \Phi(1_i, 1/2) - \Phi(0_i, 1/2) \quad (2)$$

B. Probabilistic Importance

The probabilistic importance describes the rate the uncertainty of system unreliability changes to the uncertainty in unreliability of component i changes. It can be defined as :

$$I(i \mid t) = \frac{\partial \Phi(q(t))}{\partial q_i(t)}, i = 1,2,...,n \quad (3)$$

Where:

$q_i(t)$: The unreliability of the component i at time t.

$q(t) = (q_1(t), q_2(t), ..., q_n(t))$: The unreliability vector for components in system.

$\Phi(q(t))$: The system unreliability at time t.

n : Total number of components in system.

By the total Probabilistic formula, the system unreliability can be denoted as follows (to write easy, from then on, time t is omitted):

$$\Phi(q) = (1 - q_i) \cdot \Phi(0_i, q) + q_i \cdot \Phi(1_i, q) \quad (4)$$

The probabilistic importance of the component i at time t is given as :

$$I(i \mid t) = \frac{\partial \Phi}{\partial q_i} = \Phi(1_i, q) - \Phi(0_i, q) \quad (5)$$

III. BINARY DECISION DIAGRAM METHOD

A. Binary Decision Diagram

The BDD is a graph description of the Boolean Function; it is a directed acyclic graph. As shown in Figure.1, the BDD is composed of terminal vertices, non-terminal vertices, and branches.

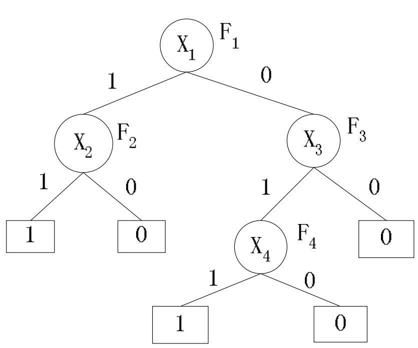

Figure 1. The binary decision diagram

Terminal vertex (Rectangle): The state of the top event in the fault tree or the state of the system, with only two states, either a 1 state which corresponds to a system failure, or a 0 state which corresponds to a system success.

Non-terminal vertex (Circle): The state of the basic event (i.e., component) in the fault tree.

Branch (Line): The state of the father vertex (i.e., non-terminal vertex) corresponding to the branch in fault tree. All the left branches leaving a vertex are 1 branches (component failure occurs) and all the right branches are the 0 branches (component functional);

Note: X_i is the index of the basic event corresponding to the non-terminal vertex; different non-terminal vertices may have the same basic event (X_i). F_i is the index of the non-terminal vertex; different non-terminal vertices must have the different F_i [7], [8].

B. Convert the Fault Tree into BDD

To compute the importance of the component by BDD method, first draws the fault tree, then converts the fault tree into BDD by If-Then-Else (ite) structure. The ite structure, which derives from Shannon's formula, is a strong tool converting the fault tree to BDD, and can be written as:

$$ite(X, M_1, M_2) = X \cdot M_1 + \overline{X} \cdot M_2 \quad (6)$$

A fault tree can be converted into a BDD by the ite structure through the follow steps:

(1) Transform the general fault tree to a regular fault tree which only contains the basic logic gate (AND, OR, NO). When the middle levels have the NO gate, eliminate the NO gate by the De.Morgan rule.

(2) From bottom to up in fault tree, replace the upstream middle events by downstream middle events or basic events through the logic operation or the ite structure.

(3) Calculate the ite structure of the top event in the fault tree and create the BDD.

To obtain the ite structure for each logic gate in the fault tree, the following relations are considered:

Let J= ite (X, M_1, M_2), H= ite (Y, N_1, N_2)

Taking X<Y, then:

J<op>H= ite (X, M_1<op>H, M_2<op>H)

Taking X=Y, then:

J<op>H= ite (X, M_1<op>N_1, M_2<op>N_2)

Where <op> correspond to the Boolean operation of the logic gate in the fault tree. For an AND gate <op> will be the dot or product symbol and for an OR gate <op> will be the addition symbol. X<Y means the basic event ordering permutation[9], [10].

C. Project Application

The fault tree as shown in Figure. 2 depicts the engine failure in one type of the airplane caused by the oil filter jammed occurring.

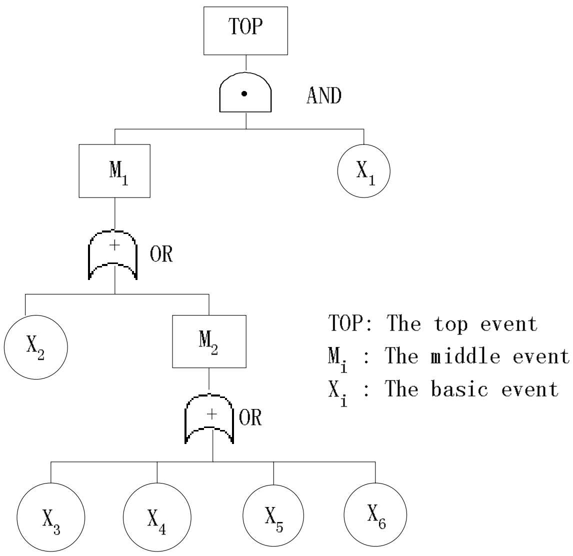

Figure 2. The fault tree of the oil filter jamming cause the engine failure

Where:

TOP – The engine fault caused by the oil filter jammed.

M1 –The oil pressure difference > 0.35Mpa, the oil pressure caution light does not give out the warning sign.

M2 –The oil pressure difference fault in oil pressure caution system.

X1 – The oil filter jammed fault.

X2 – The power system fault.

X3 – The caution light fault.

X4 – The oil filter pressure difference switch fault.

X5 – The cable group fault.

X6 – The oil pipe group fault.

As the oil filter is jammed and the oil pressure caution light does not give out the warning sign, the un-filtrated oil will go on through the shaft bearing and cause the blowpipe to be jammed, result in a serious casualty. To improve the reliability of the engine in the airplane, we must make ensure of the component that will be considered as the emphases, it is necessary to analysis the importance of the component in the oil filter and the oil pressure caution system.

Suppose the basic events ordering permutation in Figure.2 is $X_1<X_2<X_3<X_4<X_5<X_6$ (the ordering of basic events will determine the size and the complexity of the result BDD, how to use an appropriate ordering to produce an efficient result diagram, many foreign scholars advise some good ways). Then the BDD calculations for the fault tree in Figure.2 are follows:

$X_i = ite\,(X_i, 1, 0)$, and $i = 1, 2\ldots 6$

$M_2 = X_3+X_4+X_5+X_6$

$M_1 = X_2+M_2$

TOP= $X_1 \cdot M_1$

$\Rightarrow$TOP= $ite\,\{X_1, [ite\,(X_2, 1, ite\,(X_3, 1, ite\,(X_4, 1, ite\,(X_5, 1, ite\,(X_6, 1, 0)))))], 0\}$

By the *ite* structure of the top event, the result BDD is shown in Figure.3.

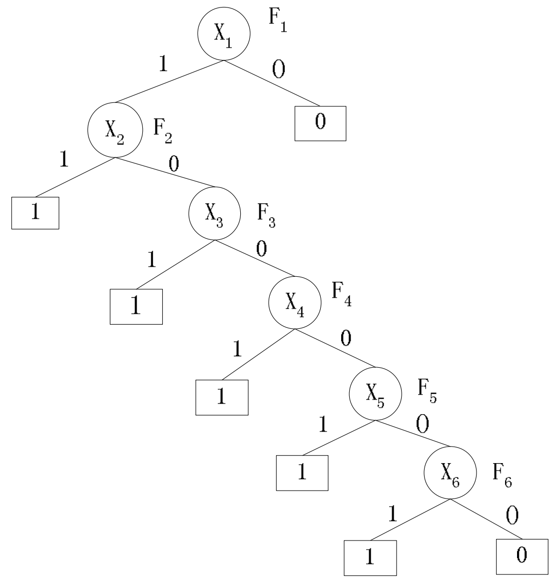

Figure 3. The BDD of the oil filter jamming cause the engine failure

D. Importance Measure by the BDD Method

For a commonly considering, let X_i is a basic event that locates in some nodes of the BDD, as show in Figure. 4.

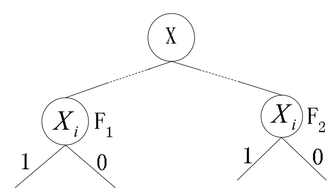

Figure 4. Basic event X_i

Introducing a criticality function $G_i(q)$ for each component, this function 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 to failed state, i.e., the probability that the system fails only if component i fails. Therefore:

$$G_i(q) = \Phi(1_i, q) - \Phi(0_i, q) \tag{7}$$

By the characteristic of the BDD, we can evaluate each of the two terms $\Phi(1_i, q)$ and $\Phi(0_i, q)$ for basic event X_i in Figure.4:

$$\Phi(1_i,q)=\sum_m\left\{p_{rxi}(q)\cdot p^1_{oxi}(q)\right\}+Z(q)$$
$$\Phi(0_i,q)=\sum_m\left\{p_{rxi}(q)\cdot p^0_{oxi}(q)\right\}+Z(q) \tag{8}$$

Where:

m : All the nodes for variable X_i on the BDD.

$p_{rxi}(q)$: The probability of the path section from the root node to node X_i.

$p^1_{oxi}(q)$: The probability of the path section from node X_i to the terminal 1 node after 1 branch from node X_i.

$p^0_{oxi}(q)$: The probability of the path section from node X_i to the terminal 1 node after 0 branches from node X_i.

$Z(q)$: The probability of the path from the root to the terminal 1 node that do not go through a node for variable X_i.

Therefore:

$$\begin{aligned}G_i(q)&=\Phi(1_i,q)-\Phi(0_i,q)\\&=\sum_m\{p_{rxi}(q)[p^1_{oxi}(q)-p^0_{oxi}(q)]\}\end{aligned} \tag{9}$$

By the criticality function $G_i(q)$ of the basic event *Xi* in formula (9), the structure importance and the probabilistic importance can be calculated, and the result as shown in appendix table 1-3. From the result, we can get the conclusion: the most important component is X_1 (the oil filter jammed fault), the most un-important component is X_3 (the caution light fault).

IV. CONCLUSION

The tables (Ⅰ, Ⅱ, Ⅲ) show that the result calculated by the BDD method, it is the same as the result by the conventional FTA method. However, the BDD method has two advantages than the FTA method:

TABLE I. THE *ite* STRUCTURE FOR EACH COMPONENT IN FIGURE.3

Node label	Basic event	unreliability for each component i at time t	1 branch pointer	0 branch pointer
F_1	X_1	0.01	F_2	0
F_2	X_2	0.15	1	F_3
F_3	X_3	0.05	1	F_4
F_4	X_4	0.12	1	F_5
F_5	X_5	0.08	1	F_6
F_6	X_6	0.10	1	0

First, the conventional top-down and bottom- up techniques applied in FTA can lead too many redundant cut sets and calculating exact top event probability can be impossible. While the BDD method implements the importance measure with high speed and gives significant savings in the computational efficiency. For example, if the number of the component in system is n , the complexity of the BDD algorithm is $O(\sqrt{n}\cdot\log_2^n)$ [12], which is much quicker than the conventional FTA method.

Second, the conventional FTA represents the system failure in a mode of the Boolean failure logic equation, which cannot be used in the computer easily. On the contrary, the BDD Method provides an alternative mathematical form, which can be implanted on computer easily and lends itself to manipulation. Also the BDD produces an exact quantified result.

TABLE II. THE STRUCTURAL IMPORTANCE (I_i^Φ) FOR EACH COMPONENT IN FIG.3

Node label	Basic event	$p^1_{oxi}(q)$	$p^0_{oxi}(q)$	$p_{rxi}(q)$	I_i^Φ
F_1	X_1	0.96875	0.0	1.0	0.96875
F_2	X_2	1.0	0.9375	0.5000	0.03125
F_3	X_3	1.0	0.8750	0.2500	0.03125
F_4	X_4	1.0	0.7500	0.1250	0.03125
F_5	X_5	1.0	0.5000	0.0625	0.03125
F_6	X_6	1.0	0.0	0.03125	0.03125

TABLE III. THE PROBABILISTIC IMPORTANCE ($I(A\mid t)$) FOR EACH COMPONENT IN FIG.3

Node label	Basic event	$p^1_{oxi}(q)$	$p^0_{oxi}(q)$	$p_{rxi}(q)$	$I(A\mid t)$
F_1	X_1	0.4116	0.0	1.0	0.4116
F_2	X_2	1.0	0.3078	0.01	0.006922
F_3	X_3	1.0	0.2714	0.0085	0.006193
F_4	X_4	1.0	0.1720	0.008075	0.006686
F_5	X_5	1.0	0.1000	0.007106	0.006395
F_6	X_6	1.0	0.0	0.006538	0.006538

REFERENCES

[1] A. Ranzy, New Algorithms for fault tree analysis, Reliability Engineering and system safety. Vol 40, 1993, pp203-211

[2] A. Rauzy, Dutuit. Y.Exact and Truncated, *Computations of Prime Implicants of Coh erent and Non-Coherent Fault Tree with Aralia*. Reliability Engineering and System Safety,1997, 58:127～144.

[3] Xinyu Zhang, Hairong Sun, *A BDD_Based Algorithm for Reliability Analysis of Phased Mission Systems*, IEEE Trans on Reliability, Vol, 46, No.1, 1999, pp: 50-60.

[4] Shengkui Zeng, Tingdi Zhao, Jianguo Zhang. *The system reliability design and analysis*. Beijing: China, Beihang University Publishing Company 2001.7.

[5] Lambert HE. *Masure of Importance of Events and Cut Sets in Fault Trees*. Reliability and Fault Tree Analysis: Theoretical and Applied Aspects of System Reliability and Safety Assessment, SIAM, Philadelphia, 1975 pp: 77-100.

[6] Yong. Ou, Joanne. Bechta, Dugan. *Sensitivity Analysis of Modular Dynamic Fault Tree*. Master's thesis, University of Virginia Department of Electrical engineering. 1999.

[7] A. Rauzy. et. al., *Computation of Prime Implicates of a Fault Tree within Aralia*. Proceedings of ESrel'95 Conference, June, 1995 pp: 190-202.

[8] R. M. Sinnamon, J. D. Andrews. *New Approaches to Evaluating Fault Trees*. Proceeding of Esrel'95 Conference, June, 1995, pp: 241-254.

[9] Cabodi. G, Camurati. P, Quer. S. *Improving the efficiency of BDD-based operators by means of partitioning*. Computer-Aided Design of Integrated Circuits and Systems, IEEE, Volume: 18, Issue: 5, May 1999 Pages: 545 – 556.

[10] Hung. W. N. N, Xiaoyu Song, Aboulhamid, E. M. Driscoll. M. A. *BDD minimization by scatter search*. Computer-Aided Design of Integrated Circuits and Systems, IEEE, Volume: 21, Issue: 8, Aug. 2002 Pages: 974 – 979.