European Association for the Development of Renewable Energies, Environment and Power Quality (EA4EPQ)

International Conference on Renewable Energies and Power Quality (ICREPQ'12)
Santiago de Compostela (Spain), 28th to 30th March, 2012

WIND TURBINES MAINTENANCE MANAGEMENT BASED ON FTA AND BDD

F.P. García Márquez[1], J.M. Pinar Pérez[1] M. Papaelias[2] and R. Ruiz Hermosa[1]

[1] Ingenium Research Group, E.T.S.I.I., Castilla-La Mancha University
Campus Universitario s/n, 13071 Ciudad Real (Spain)
Phone Nº:+34 926 295300, e-mails: {FaustoPedro.Garcia, JesusMaria.Pinar}@uclm.es, raul.ruiz.hermosa@gmail.com

[2] Engineering and Physical Sciences College, University of Birmingham, UK
m.papaelias@bham.ac.uk

Abstract. The condition of a wind turbine will depend on different elements that compound it. The relationships cause-effects of the different faults can be analysed qualitatively by Fault Tree Analysis (FTA). FTA is a graphical representation of logical relationships between events, where each event has a fault probability associated. In this paper is proposed the quantitatively study of the FTA via Binary Decision Diagram (BDD). BDD is a method which determines the output value of the function by examining the inputs. The BDD method does not analyse the FTA directly, but converts the tree to the Boolean equations that will provide the fault probability of the top event. This conversion presents several problems, where the variable ordering scheme chosen for the construction of the BDD has a crucial effect on its resulting size. It is solved in this paper employing the Level, Top-down-Left-Right, AND, Depth First Search and Breadth-First Search methods for ranking the events (or vertices), and a comparative analysis is done.

Key words

Fault Tree Analysis, Binary Diagram Decision, Wind Turbines, Maintenance.

1. Introduction

An identification of potentially hazardous events in a wind turbine is necessary for assessment of their consequences and frequency of occurrence. It will lead to reduce costs, and to increase the reliability, availability, maintainability and safety (RAMS) of the wind turbines. In this research work is proposed a Fault Tree Analysis (FTA) as a graphical representation of logical relation-ships between the elements that compound the wind turbines. Complex systems analysis may produce thousands of combinations of events (cut sets) that can cause the system failure. The determination of these cut sets can be a large and time-consuming process even on the modern computers, and if the fault tree has many cut sets, the determination of the exact top event probability also requires lengthy calculations. For many complex fault trees this requirement may be beyond the capability of the available computers. As a consequence, approximation techniques have been introduced with a loss of accuracy [7]. In this paper is proposed the Binary Diagram Decision for obtaining the Boolean expression to solve the probability of the top event in the FTA.

2. Binary Diagram Decisions

Binary Decision Diagrams (BDDs), as a data structure that represents the Boolean functions, were introduced by Lee (1959) [5]. BDD provides a new alternative to the traditional cut-set based approach for FTA that leads to the determination of the output value of the function through the examination of the values of the inputs.

A BDD is a directed acyclic graph (**V, N**), with vertex set **V** and index set **N**. Vertex set contains two types of vertices. On the one hand, a terminal vertex has as attribute a value: value(v) $\in$ {0,1}, where a 1 state, that corresponds to system unsuccessful, or a 0 state which corresponds to a system success. All the paths that have 1 state provide the cut sets of the fault tree. On the other hand, a non terminal vertex v has as attributes an argument index(v) $\in$ **N** {0,1,…,n} and two descendants, low(v) and high(v) $\in$ **V**, that are connected by a branch. Each vertex has a vertex 0 branch that represents a non occurrence basic event, or 1 branch that represents an occurrence basic event. For any non-terminal vertex v, if low(v) is also non-terminal, then index(v) < index(low(v)), and if high(v) is non-terminal, then index(v) < index(high(v)).

A BDD has a root vertex v that leads to denote a function f_v which is defined recursively as: Firstly, if v is a terminal vertex and value(v) = 1, then f_v = 1. In other case, when value(v) = 0 then f_v = 0; Secondly, if v is a non terminal vertex with index(v) = 1, then f_v will be:

$$f_v(x_1,\ldots,x_n) = \boldsymbol{x_i}\cdot f_{low(v)}(x_1,\ldots,x_n) + x_i \cdot f_{high(v)}(x_1,\ldots,x_n)$$

3. Conversion from FTA to BDD

The following template conversion method is used for obtaining the BDD from the FTA [4]. Then the level of unreliability can be determined from the BDD easily.

Let A be a vertex set as A = A(A_1, ..., A_n). If A_1, ..., A_m are the A descendant vertices, then:

$$\text{index}(A(A_1, \ldots, A_n)) = \min(\text{index}(G_i)), \text{ where } 1 \le i \le n.$$

Operating with the Boolean variables, e.g. with operations as If-Then-Else (ite), it is possible to apply the following rules in order to obtain the BDD: Get-rid-of formula; Expansion formula; Absorption formula; Changed-order formula.

The size of the BDD will depend on the index(G_i) and is crucial in order to reduce the size, and thus the computational time to solve the BDD. There are different methods, and any of them will be more adequate to use according to the problem structure, number of variables, etc. In this paper has been considered the Level, Top-down-Left-Right, AND, Depth First Search and Breadth-First Search methods for listing the events (or vertices) A_i, and a comparative analysis has been done.

4. Rankings for Events

The level in any event is understood as the number of the gates that has higher up the tree until the top event. The "level" method creates the ranking of the events regarding to the level of them. In case that two or more events have the same level, the event will have highest priority if it appears early in the tree [6].

Top-down-left-right (TDLR) method generates a ranking of the events by ordering them from the original fault tree structure in a top-down and then left-right manner [1]. In other words, the listing of the events is initialized, at each level, in a left to right path and basic events that are found are added to the ordering list. In case that any event is encountered and it is already located higher up the tree and has therefore already been incorporated in the list, then it is ignored.

Xie *et al.* (2000) suggest by the AND criterion that the importance of the basic element is based on the "and" gates that there are between the *k* element and the top element, because in FTA the "and" gates imply that there are redundancies in the system. Consequently, basic events under an "and" gate can be viewed as less important because it is independent to other basic events occurring for the intermediate events [7].

The depth first search (DFS) method goes from top to down of the tree and each sub-tree on the left. It is a non-recursive implementation and all freshly expanded nodes are added as last-input last-output process [2].

The breadth-first search (BFS) algorithm begins ordering all the child events obtained expanding from the standpoint by the first-input first-output procedure (FIFO). The events do not considered are added in a queue list named "open". It is recalled "closed" list when it is studied [3].

5. Case Studies

The FTA showed in Fig. 1 has been taken as an example case study, where it is explained in detail how are obtained the associated BDD and the fault probability. It has two multiple occurring events that occurs more than one place in the FTA, B and C, also known as a redundant or repeated event.

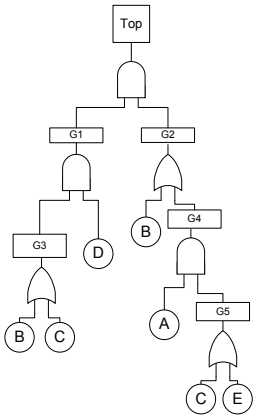

Fig. 1. FTA with some repeat events.

The ranking obtained has been done by the methods of Level, TDLR, AND, DFS and BFS, and the lists are showed in Table I.

Table I. Rankings obtained for the Case Study

Success	LEVEL	TDLR	*AND*	DFS	BFS
A	3	4	*4*	4	4
B	3,2	2	*1*	1	2
C	3,4	3	*3*	2	3
D	2	1	*2*	3	1
E	*4*	*5*	*5*	*5*	*5*

The logic equations of the gate outputs are given by: G5=C+E; G4=A+G5; G3= B+ C; G2= B+ G4; G1=D+G3.

The BDD associated to the FTA (Fig. 1) is shown in Fig. 2.

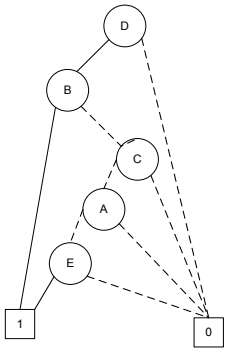

Fig. 2. BDD of the case study

From BDD showed in Fig. 2 can be obtained the probability of the top event, *Q*, as:

$$Q = q_D q_B + q_D (1 - q_B) q_C q_A q_E$$

The probabilities *q* of successes A, B, C, D and E, and the probability of the system, *Q*, are shown in Table II. The first row considers the same fault probability of each success. The fault probability of each success has been multiplied by two in following rows, where the last column shows the success whose probability has been modified.

The most important events for the size of the BDD are B and D. B is a multiple occurring event, but D is a simple occurring event. As it is showed in Table I, the TDLR and BFS and AND methods find these events in the firsts two positions of the rankings, but only the AND provided B as first in the ranking, where the other methods, that find the same solutions for every events, find D first in the ranking.

Table II. Probabilities for A, B, C, D and E events

q_A	q_B	q_C	q_D	q_E	Q	Event
0,001	0,001	0,001	0,001	0,001	0,000001	
0,002	0,001	0,001	0,001	0,001	0,000001	A
0,001	**0,002**	0,001	0,001	0,001	0,000002	B
0,001	0,001	**0,002**	0,001	0,001	0,000001	C
0,001	0,001	0,001	**0,002**	0,001	0,000002	D
0,001	0,001	0,001	0,001	**0,002**	0,000001	E

A set of fault trees has been considered in order to test the ranking obtained by the methods aforementioned. They are presented in Table III. Different sizes of the events and structures, considering structures as number of "and" and "or" gates, and levels, has been considered.

Table III. Fault Tree case studies

FAULT TREE	Size	and	or	Levels
a	4	2	2	2
b	5	3	3	3
c	6	3	3	3
d	7	3	3	2
e	8	3	3	2
f	11	5	5	4
g	12	2	10	7
h	12	3	10	3
i	19	6	8	3
k	25	6	16	12
l	17	8	9	5

6. Results

The effectiveness of the Level, TDLR, AND, DFS and BFS methods has been done regarding to the cut sets number obtained by the BDD. If the size of cut sets increases, then the computational time required for calculating the *Q* probability of the top event will rise.

The numbers of cut sets of the fault trees, described in Table III, obtained by the And, Level, BFS, DFS and TDLR methods are shown in Fig. 3. To reduce the number of cut sets depends on the method employed for listing and the fault tree. BFS provides generally bad results in most of the cases, especially when the fault tree has a high number of events, levels and "or" and "and" gates. Otherwise the Level and AND methods generate the ordering of the events with a reduced number of cut sets. The conclusions regarding to Level, DFS and TDLR methods should be studied for each fault tree.

The results of the computational time required for solving the problems are proportional to the number of cut sets showed in Fig. 3, and therefore the conclusions aforementioned.

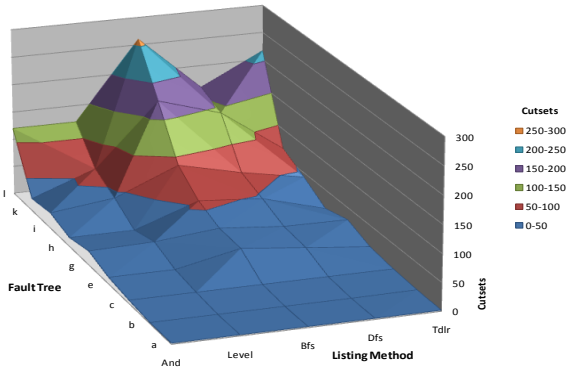

Fig. 3. Numbers of cut sets given by AND, Level, BFS, DFS and TDLR methods

7. Conclusions

To solve quantitatively a fault tree analysis (FTA) generates a large number of operations and therefore a high computational cost. In this paper is suggested the binary diagram decision (BDD) for obtaining the Boolean expression to solve the probability of the FTA top event. The cut sets generated by BDD will depend on the events ordering.

The "Level", "Top-Down-Left-Right", "AND", "Depth-First Search" and "Breadth-First Search" methods has been considered for listing the events, and a comparative analysis has been done. The Level and AND methods create the listing of the events that provide a reduced number of cut sets. The Level, Depth-First Search and Top-down-Left-Right methods should be studied for each fault tree. Finally the Breadth-First Search is the ordering method that provides a higher cut sets number.

Acknowledgements

This work has been supported by the European Project NIMO (Ref.: FP7-ENERGY-2008-TREN-1: 239462).

References

[1] Bartlett L.M. (2003). Progression of the binary decision diagram conversion methods. Proceedings of the 21st International System Safety Conference, August 4-8, 2003, Ottawa, Westin Hotel, pp 116-125.

[2] Cormen T. H., Leiserson C. E., Rivest R .L. and Stein C. (2001). Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill. ISBN 0-262-03293-7. Section 22.3: Depth-first search, pp.540–549.

[3] Jensen, R., and Veloso, M. M. (2000). OBDD-based universal planning for synchronized agents in non-deterministic domains. Journal of Artificial Intelligence Research 13:189–226.

[4] Jinglun Z., and Quan S. (1998). Reliability Analysis Based on Binary Decision Diagrams. Journal of Quality in Maintenance Engineering, vol. 4, num. 2, pp. 150-161.

[5] Lee C. Y. (1959). Representation of switching circuits by binary decision diagrams. Bell System Technology, vol. 38, pp. 985-999.

[6] Malik S., Wang A.R., Brayton R.K., Vincentelli A. S. (1988). Logic Verification using Binary Decision Diagrams in Logic Synthesis Environment. In Proceedings of the IEEE International Conference on Computer Aided Design, ICCAD'88. Santa Clara CA, USA, pages 6–9.

[7] Xie M., Tan K.C., Goh K.H. and Huang X.R. (2000). Optimum Prioritisation and Resource Allocation Based on Fault Tree Analysis. International Journal of Quality & Reliability Management, vol. 17, Nº 2, pp. 189-199.