Important Fault Tree Characteristics For Efficient BDD Construction

L.M.Bartlett

Loughborough University

Loughborough

Leicestershire

LE11 3TU

Keywords - Fault tree analysis, Binary Decision Diagrams, Neural Networks, Variable Ordering, Jacobian matrix.

Summary & Conclusions - The Binary Decision Diagram (BDD) methodology is the latest approach used to improve the analysis of the fault tree diagram, which gives a qualitative and quantitative assessment of specified risks. To convert the fault tree into the necessary BDD format requires the basic events of the tree to be placed in an ordering. The ordering of the basic events is critical to the resulting size of the BDD, and ultimately affects the performance and benefits of this technique. A number of heuristic approaches have been developed to produce an optimal ordering permutation for a specific tree, however, they do not always yield a minimal BDD structure for all trees. Latest research considers a neural network approach used to select the 'best' ordering permutation from a given set of alternatives. To use this approach characteristics are taken from the fault tree as guidelines to selection of the appropriate ordering permutation. This paper looks at a new method of using the Jacobian matrix to choose the most desired characteristics from the fault tree, which will aid the neural network selection procedure.

1. INTRODUCTION

The introduction of the Binary Decision Diagram[1-6] approach has helped overcome the numerical and efficiency limitations of the fault tree analysis technique when dealing with large fault tree structures. When transformed to the necessary format the BDD analysis procedure is qualitatively more efficient and has greater precision quantitatively. The transformation process is sometimes where problems lie, in that the resulting diagram is not minimal in size. To make the conversion the basic events of the fault tree need to be taken in a specified order. This ordering is crucial to the size of the resulting BDD, where a good ordering can result in a very efficient analysis and a poor ordering can lead to problems.

Within the literature there are a number of possible ordering heuristics to convert the fault tree structure[7-11]. Unfortunately, there is not a single heuristic that will guarantee that the end result of the conversion process will be minimal. However, it is very likely that one of the heuristics available will produce the BDD required, the problem is finding it. This is where pattern recognition approaches have been used. Genetic algorithms[12] and neural network based approaches[13,14] have been used to select an ordering heuristic from a set of alternatives based on specific characteristics of the fault tree. The neural network approach has been the most successful predicting fourteen out of twenty fault trees from a test set with the correct ordering heuristic to produce a minimal BDD. The method used eleven characteristics from the fault tree to choose the appropriate ordering heuristic from a set of six for the conversion process. The initial research showed the neural network to be a useful technique but more work was needed to improve its predictive capabilities. One of the areas for possible improvement is the inputs to the problem, namely the eleven fault tree characteristics. Previously intuitively chosen, this paper looks at a new, more precise mathematical method to calculate the sensitivity of the outputs (ordering heuristics) to the inputs (fault tree characteristics). This new method uses the Jacobian matrix to indicate which of the eleven characteristics has a strong influence in determining the choice of ordering heuristic. To determine the credibility of the method important influential characteristics highlighted in the research have been used to retrain the neural network to see if the same or a better predictive capability can be achieved.

2. THE NEURAL NETWORK APPROACH

2.1 Overview of Neural Network Approach

The multi-layer perceptron approach[15] which has been used in previous research[13,14] and in this continuing research is a method of identifying patterns. The pattern in this ordering problem is between the fault tree structure and the ordering heuristic that will produce a minimal BDD. The network comprises a layer of input nodes which correspond to the fault tree characteristics, a layer of outputs which refer to the ordering heuristic choices, and a layer of hidden nodes. Each of the layers are connected to the previous layer by means of weights which govern the pattern recognition potential of the network, and are set during training. The outputs of the nodes at differing layers of the network are created by taking the sum of the product of the inputs and the weights attached to each input, and applying a non-linear function to the result. For the hidden layer, the output for node j (a_j) is given by equation (1), where w_{ji} refers to the weight connecting hidden node j to input node i, and x_i corresponds to the input node value. The application of a non-linear function, usually the sigmoidal function, produces the activated output z_j.

$$a_j = \sum_i w_{ji} x_i \tag{1}$$

Equation (2) is used for the final layer outputs, where w_{kj} refers to the weight connecting output node k to hidden node j. Again the activated output, z_k is produced by applying another activation function which can be the same or different than used for the hidden layer nodes.

$$a_k = \sum_j w_{kj} z_j \tag{2}$$

The method applies an error-back propagation technique whereby the weights are altered according to an error correction rule that is carried back through the network from the output layer to the input layer. This methodology serves to alter the weights in such a way as to move the result produced by the network toward the desired result. The network is trained on a large data set of examples, where the pattern is learned, and then can be used in a predictive capacity for previously unseen data.

2.2 Methodology for Ordering Problem

The difficulty in the neural network approach is in correctly modelling the problem. Previous work[14] has chosen fault tree attributes intuitively. Eleven characteristics were selected to represent the fault tree structure and six ordering heuristic preferences were used as the selection set.

The characteristics that were chosen to represent the fault tree structure are: the percentage of AND gates, percentage of different events repeated, percentage of total events repeated, top gate type, number of outputs from the top gate, number of levels in the tree, number of basic events, maximum number of gates in any level, number of gates with just event or gate only inputs, and highest number of repeated events. A full description and reasons for inclusion can be found in reference [14].

The ordering heuristics chosen to be the alternatives used for selection are:

- Top-down, left-right approach;
- Depth-first approach;
- Priority depth-first approach;

The remaining three heuristics are repeated event versions of each of the above. The details of each of these ordering heuristics can be found in reference [16].

To train and test the neural network a set of examples was required. Fault tree structures were used from industry and randomly generated using a computer program. All trees were analysed for the chosen eleven characteristics and best ordering heuristic alternative (using the number of nodes in the diagram).

To evaluate the performance of the neural network a test set of data was produced with different tree structures and known best ordering heuristics. The number of correct ordering heuristic preferences predicted by the network quantified the performance.

2.3 Problems

Conclusions from the research indicated that the neural network approach is a novel technique to help in this ordering problem of trying to guarantee a minimal BDD in the conversion process. Results from the research showed that using the neural network methodology which allows for the selection of an ordering heuristic from a set to produce a minimal BDD, not just a BDD, had a predictive capability of seventy percent for the test set used. The remaining thirty percent of responses were variable. Although this technique proved to be considerably better than using any single heuristic alone, there is still room for improvement to increase the predictive potential. It is suggested[14] that the characteristics of the fault tree be scrutinised as to their relevance in determining the best ordering heuristic to use. If the inputs of the problem do not contain the necessary information to learn the required pattern then applying any pattern recognition technique will be unsuccessful, thus the inputs need to be selected carefully. The aim of the current paper is to make this selection using the Jacobian method, which will be explained subsequently, to highlight the important characteristics of a fault tree in the conversion process to a BDD.

3. USING THE JACOBIAN METHOD

3.1 The General Approach

The back propagation technique used in the multi-layer perceptron approach can be applied to the calculation of other derivatives. In particular it can be used to look at the sensitivity of the outputs with respect to the inputs, which is done by evaluating the Jacobian matrix. The sensitivity of output k to input i is defined by the notation J_{ki}. The elements of this matrix are given by the derivatives of the network outputs with respect to the inputs:

$$J_{ki} = \frac{\partial y_k}{\partial x_i}$$

where y_k is the output and x_i is the input.

The general procedure can be described in the following steps:

i) Apply the input pattern for which the Jacobian matrix is to be found and forward propogate to obtain the activations of all the hidden nodes (z_j) using equation (3) and output units (z_k) in the network using equation (4).

$$z_j = \frac{1}{\exp\left(-\sum_i w_{ji} x_i\right)} \tag{3}$$

$$z_k = \frac{1}{\exp\left(-\sum_j w_{kj} z_j\right)} \tag{4}$$

ii) Next the Jacobian for each output needs to be calculated using equation (5). The Jacobian values are calculated by recursively passing back through the network, summing initially over l (part 1), which corresponds to all the output nodes to which the hidden node j in the first summation (part 2) is connected. Then finally considering the connections between the hidden and input layer using part 2 of equation 5.

$$
\begin{aligned}
J_{ki} &= \frac{\partial y_k}{\partial x_i} = \sum_j \frac{\partial y_k}{\partial a_j}.\frac{\partial a_j}{\partial x_i} = \sum_j w_{ji} \frac{\partial y_k}{\partial a_j} \\
&= \sum_j w_{ji} \sum_l \frac{\partial y_k}{\partial a_l} \frac{\partial a_l}{\partial a_j} = \sum_j w_{ji} \; z'_j \sum_l w_{lj} \frac{\partial y_k}{\partial a_l} \qquad (5) \\
&= \underbrace{\sum_j w_{ji} \; z'_j}_{Part\ 2} \underbrace{\sum_l w_{lj} \; z'_k \; \delta_{kl}}_{Part\ 1}
\end{aligned}
$$

The δ_{kl} term in part 1 is set to 1 if output *k* equals node *l* in the output layer, and 0 otherwise. The terms z'_j and z'_k in part 2 and 1 respectively correspond to the differentials of equations (3) and (4).

iii) Steps (i) and (ii) are repeated for all output nodes *k* with respect to all possible inputs *i*.

Each of the derivatives is calculated with all the other inputs held fixed. As the trained network represents a non-linear mapping the elements of the Jacobian matrix will not be constants but will vary depending on the input vector being evaluated. Thus, the Jacobian must be re-evaluated for each new input vector.

3.2 Application of The Jacobian Method to Ordering Problem

Ideally the inputs to the neural network are characteristics which are influential in deciding which of the ordering heuristic options is the best. The title ‘best’ is given to the heuristic that produces the minimal BDD. To investigate which characteristics are important the sensitivity of each of the ordering heuristics with respect to each of the input characteristics has been calculated.

When calculating the sensitivity using the Jacobian method the result can be positive or negative declaring the direction of influence. To find the important characteristics, those with most influence in determining the ordering heuristic choice, the direction is not a concern but the magnitude of the influence is. Therefore, the modulus of each Jacobian value has been used. Thus, the larger the Jacobian value the larger the influence on selection.

To calculate this sensitivity the data set used to train the neural network has been used, so the sensitivity has been compared using 198 input patterns. As the Jacobian must be re-evaluated for each new input pattern, an average Jacobian value has been calculated, i.e. the outcome is the average sensitivity of ordering heuristic k to each characteristic i. If the average sensitivity of output node 2 to input node 1 ($Ave\ J_{21}$) was being calculated, for example, the following equation would be used, where the number in superscript is the input pattern number.

$$Ave\ J_{21} = \frac{J_{21}^{(1)} + J_{21}^{(2)} + J_{21}^{(3)} + \cdots + J_{21}^{(198)}}{198}$$

The results are given in Table 1 (columns 2 – 7).

The main concern is to determine whether the input characteristic is important. As a measure of this, the effect of input i has been averaged over all the outputs and over all the training patterns, as can be seen with the results in table 1 (final column). For example, if the network had three output nodes and three input nodes, the average effect of input 2 ($Ave\ J_{Input\,2}$) would be given by:

$$Ave\ J_{Input\,2} = \frac{J_{12}^{(1)} + J_{22}^{(1)} + J_{32}^{(1)} + J_{12}^{(2)} + J_{22}^{(2)} + J_{32}^{(2)} + \cdots + J_{12}^{(198)} + J_{22}^{(198)} + J_{32}^{(198)}}{(198)(3)}$$

	Scheme 1	Scheme 2	Scheme 3	Scheme 4	Scheme 5	Scheme 6	Average
Input 0	0.011888	0.036141	0.019004	0.040422	0.023307	0.012616	0.023896
Input 1	0.023687	0.007152	0.003790	0.008052	0.004668	0.002501	0.008309
Input 2	0.016529	0.011386	0.005662	0.012223	0.006915	0.003931	0.009441
Input 3	0.011705	0.025486	0.013745	0.029002	0.016887	0.008929	0.017626
Input 4	0.022219	0.030165	0.016269	0.034237	0.020044	0.010536	0.22245

Input 5	0.037601	0.022271	0.010790	0.023640	0.013117	0.007710	0.019188
Input 6	0.008122	0.008316	0.004338	0.009295	0.005329	0.002912	0.006385
Input 7	0.018904	0.029979	0.014899	0.032388	0.018151	0.010421	0.020790
Input 8	0.036213	0.069418	0.036050	0.077007	0.044152	0.024198	**0.047840**
Input 9	0.004847	0.008101	0.003996	0.008717	0.004884	0.002814	*0.005560*
Input 10	0.045928	0.007804	0.003807	0.008354	0.004639	0.002713	0.012207

Table 1. Average Jacobian values

From the results in table 1 it can be seen that although the values are not very large there is a considerable difference in their magnitude. The largest overall average value (column 8) is obtained for input 8, which corresponds to the number of gates with just event inputs, with a value of 0.047840. The smallest Jacobian value is for input 9, the number of gates with just gate inputs. By ranking the average effect of each input, as shown in table 2, it can be seen that input 8 has twice the influence of the second ranked characteristic and almost 10 times the influence of characteristic 9. The ranked results are interesting because it has highlighted certain characteristics which where previously thought influential, i.e. number of basic events, to be very low in terms of effect.

Rank	Input Number	Characteristic	Jacobian average value
1	8	Number of gates with just events inputs	**0.047840**
2	0	Percentage of AND gates	**0.023896**
3	4	Number of levels	**0.022245**
4	7	Max number of gates	**0.020790**
5	5	Number of outputs from top gate	**0.019188**
6	3	Top gate type	**0.017626**
7	10	Highest repeated event	0.012207
8	2	Percentage of total events repeated	0.009441
9	1	Percentage of different events repeated	0.008309
10	6	Number of basic events	0.006385
11	9	Number of events with gate only inputs	0.005560

Table 2: Ranked Jacobian results

Having highlighted distinct differences in the magnitude of influence for certain characteristics the next step was to determine whether just using the more influential characteristics could still predict the best ordering heuristic for a given fault tree.

A subset of the characteristics in table 2 were taken and the neural network retrained and tested to see whether the same or improved predictive results could be found.

3.3 Does It Improve The Neural Network Technique?

It was decided to take the top six characteristics from table 2, with the sixth ranked characteristic having approximately three times less influence than the top ranked characteristic, with the remaining characteristics having four or more times less influence. These six characteristics were used to retrain the neural network to see whether the prediction of the best ordering method is the same or better than the fourteen out of twenty correct responses predicted in reference [14].

The multi-layer perceptron approach was used with an input layer of 6 nodes, an output layer as the original network with 6 nodes, and one layer of hidden nodes. The best network architecture involved 5 hidden nodes. The outputs of each layer of the network were generated using equations (1) and (2), and a sigmoidal activation function was applied to both layers to produce the activated outputs. The network was tested on the same test set of fault trees as in reference [14] and each tree had a known best ordering heuristic. The predictive capability reached was fourteen out of twenty correct ordering heuristic choices. This is equal to that gained using eleven characteristics, suggesting that the remaining 5 characteristics omitted from this testing procedure provide no further information to the neural network to help in the pattern recognition potential.

4. CONCLUSIONS

Using the Jacobian method has provided an insight into the important characteristics of a fault tree in the BDD conversion process. Reducing the set of characteristics from eleven to six has yielded the same predictive potential when using the neural network technique, thus making the workings of the network more efficient for the same response. Further work to enhance the predictive potential is to investigate additional fault tree characteristics and establish their importance using the Jacobian method before the network is trained. Also the neural network selection mechanism can be improved further by extending the range of ordering heuristics used to produce the BDD, which will in turn involve subsequent analysis of the fault tree characteristics. It is clear that this is a beneficial technique to the ordering problem and can aid in further developing this neural network method and ultimately find the necessary characteristics to increase the predictive potential significantly to warrant using this approach in a commercial package.

REFERENCES

1. Akers, S.B. Binary Decision Diagrams. *IEEE Transactions on Computers*, vol. C-27, 1978, p509-516.
2. Rauzy, A. A Brief Introduction to Binary Decision Diagrams. *European Journal of Automation*, vol. 30, No. 8, 1996.
3. Rauzy, A. New algorithms for fault tree analysis. *Reliability Engineering and System Safety*, vol. 40, 1993, p203-211.
4. Sinnamon, R.M. and Andrews, J.D. Quantitative Fault Tree Analysis Using Binary Decision Diagrams. *European Journal of Automation*, vol. 30, No. 8, 1996.
5. Bryant, R.E. Graph-Based Algorithms for Boolean Function Manipulation. *IEEE Transactions on Computers*, vol. C-35, No. 8, 1986.
6. Sinnamon, R.M and Andrews, J.D. Improved Efficiency in Qualitative Fault Tree Analysis. *Advances in Reliability Technology Symposium*, Manchester, 1996.

7. Sinnamon, R.M and Andrews, J.D. Fault Tree Analysis and Binary Decision Diagrams. *Proceedings of the Reliability and Maintainability Symposium*, Las Vegas, January 1996.
8. Minato, S., Ishiura, N. and Yajima, S. On variable ordering of binary decision diagrams for the application of the multi-level logic synthesis. *Proceedings of the European Conference on Design Automation*, 1991, p50-54.
9. Bouissou, M., Bruyere, F. and Rauzy, A. BDD Based Fault-Tree Processing: A Comparison of Variable Ordering Heuristics. *Proceedings of ESREL '97 Conference*, August 1997.
10. Bouissou, M. An Ordering Heuristic for Building Binary Decision Diagrams from Fault Trees. *Proceedings of Reliability and Maintainability Symposium*, Las Vegas, 1996, p208-214.
11. Fujita, M., Fujisawa, H. and Kawato, N. Evaluation and Improvements of Boolean Comparison Method Based on Binary Decision Diagrams. *IEEE Transactions on Computer Aided Design (Conference)*, Nov 1988, p2-5.
12. L.M.Bartlett and J.D.Andrews. Efficient Basic Event Ordering Schemes for Fault Tree Analysis. *Quality and Reliability Engineering International*, Vol. 15, 1999, p95-101.
13. L.M.Bartlett and J.D.Andrews. Comparison of Two New Approaches to Variable Ordering for Binary Decision Diagrams. *Proceedings of Advances in Reliability Techniques Symposium*, Nov. 2000, Manchester.
14. L.M.Bartlett and J.D.Andrews. Selecting An Ordering Heuristic For The Fault Tree to Binary Decision Diagram Conversion Process Using Neural Networks. *IEEE Transactions on Reliability*, 51(3) , 2002, pp 344-349.
15. C.M.Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.
16. Andrews, J.D. and Bartlett, L.M. Efficient Basic Event Orderings For Binary Decision Diagrams. *Proceedings of the Annual Reliability and Maintainability Symposium*, 1998, p61-68.