

This item was submitted to Loughborough's Institutional Repository by the author and is made available under the following Creative Commons Licence conditions.



creative  
commons  
C O M M O N S D E E D

**Attribution-NonCommercial-NoDerivs 2.5**

**You are free:**

- to copy, distribute, display, and perform the work

**Under the following conditions:**

 **Attribution.** You must attribute the work in the manner specified by the author or licensor.

 **Noncommercial.** You may not use this work for commercial purposes.

 **No Derivative Works.** You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

**Your fair use and other rights are in no way affected by the above.**

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:  
<http://creativecommons.org/licenses/by-nc-nd/2.5/>



# Important Fault Tree Characteristics for Efficient BDD Construction

Author: Dr Lisa Bartlett

Presenter: Dr Suguo Du

# Background to Research

- **From a practical standpoint:**
- Commercial fault tree packages now readily available, use traditional kinetic tree theory for analysis.
- Latest development for fault tree analysis is the Binary Decision Diagram (BDD) methodology. More accurate, more efficient.
- Problems with conversion from fault tree to this BDD format. Advantages of technique can not be utilised.
- Research needed in conversion process.

# Background to Research

- **Research in conversion process:**
- Several ordering heuristics available.
- No one ordering heuristic guarantee good conversion for all fault trees.
- Latest approach is to use a selection procedure where heuristics are chosen depending on fault tree.
- Selection procedure uses neural networks.

# Overview of Research

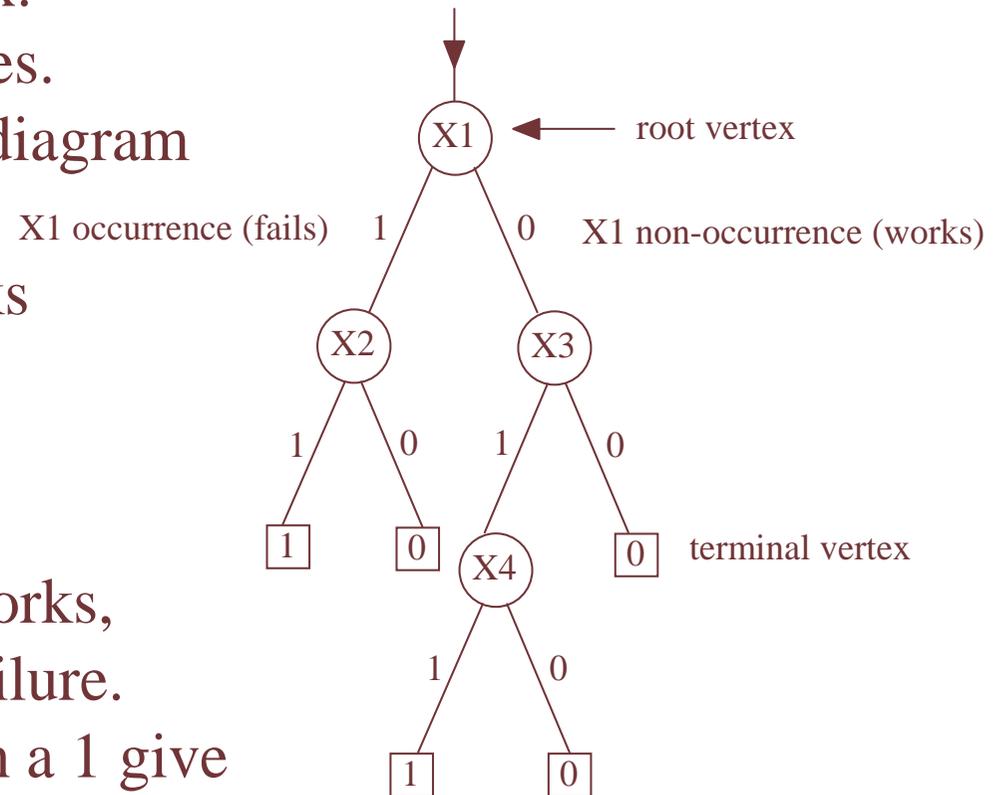
- **Research into important fault tree characteristics:**
- Neural network makes choices depending on fault tree characteristics.
- To improve performance characteristics need to be scrutinised.
- Method to analyse characteristics – Jacobian method.
- Ideal outcome to produce neural network selection procedure to guarantee BDD conversion. Hence, advantages of BDD methodology utilised.

# Overview of Presentation

- Binary Decision Diagram.
- Conversion Heuristics – Neural network approach.
- Fault tree characteristics used in selection procedure.
- New method for choosing fault tree characteristics – Jacobian method.
- Results using new method.

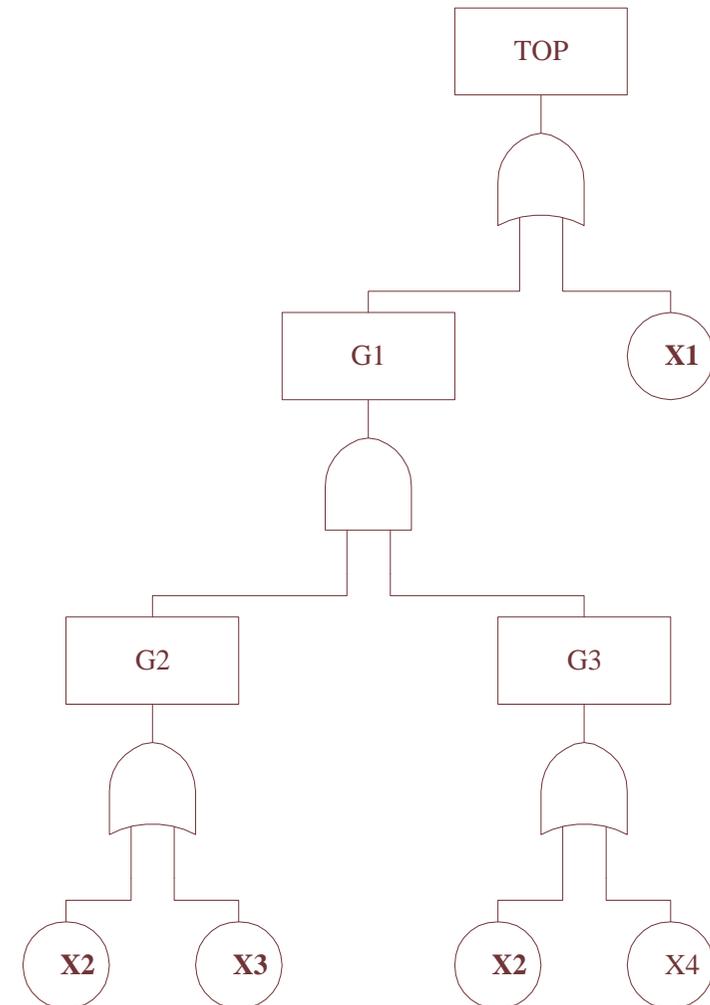
# Binary Decision Diagram

- Starts with a root vertex.
- Ends in terminal vertices.
- Each path through the diagram terminates in:
  - 0 - system works
  - 1 - system fails
- Vertex = basic event.  
Two branches:
  - 0 component works,
  - 1 component failure.
- All paths terminating in a 1 give cut sets:
- Basic events must be ordered.



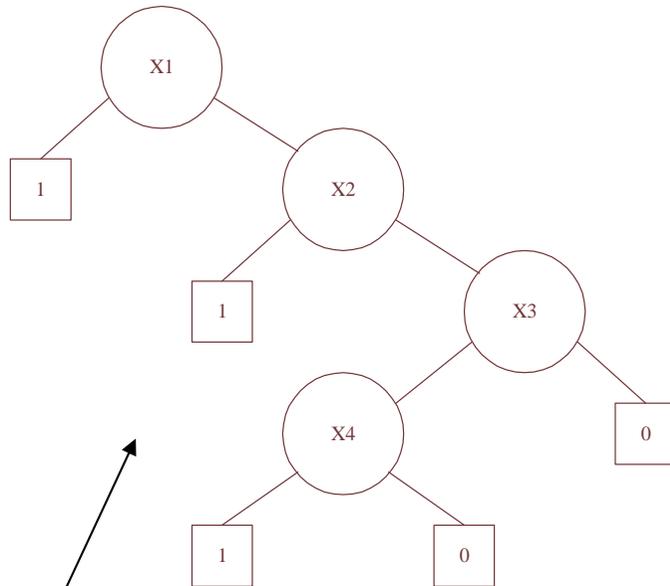
# BDD - Problems

- Fault tree conversion require basic events to be ordered.
- Many different possible orderings.
- Ordering 1:  
 $X1 < X2 < X3 < X4$
- Ordering 2:  
 $X4 < X3 < X2 < X1$
- Different sized BDD's.

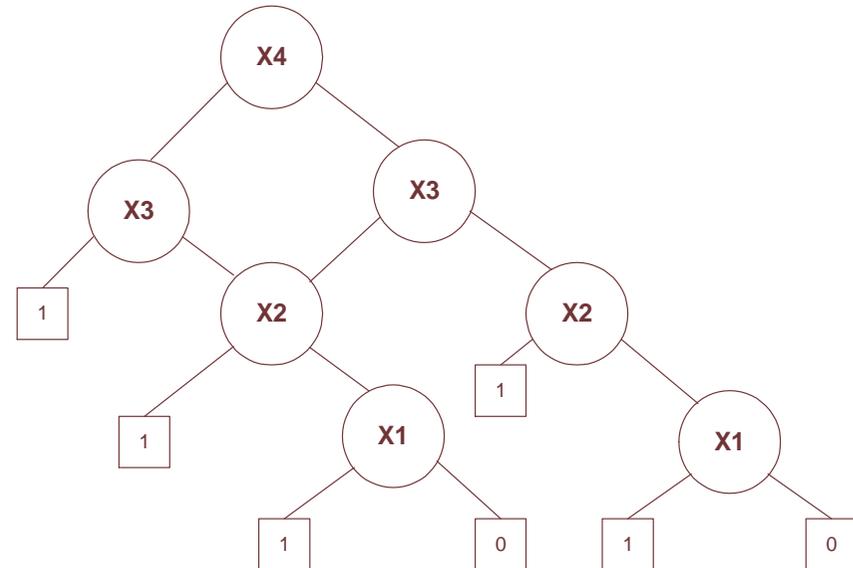


# BDD - Problems

- Ordering 1:  
 $X1 < X2 < X3 < X4$



- Ordering 2:  
 $X4 < X3 < X2 < X1$



- BDD is minimal. Four nodes only. Minimal cut sets directly, and quantification with fewest possible steps.

# Ordering Heuristics - General

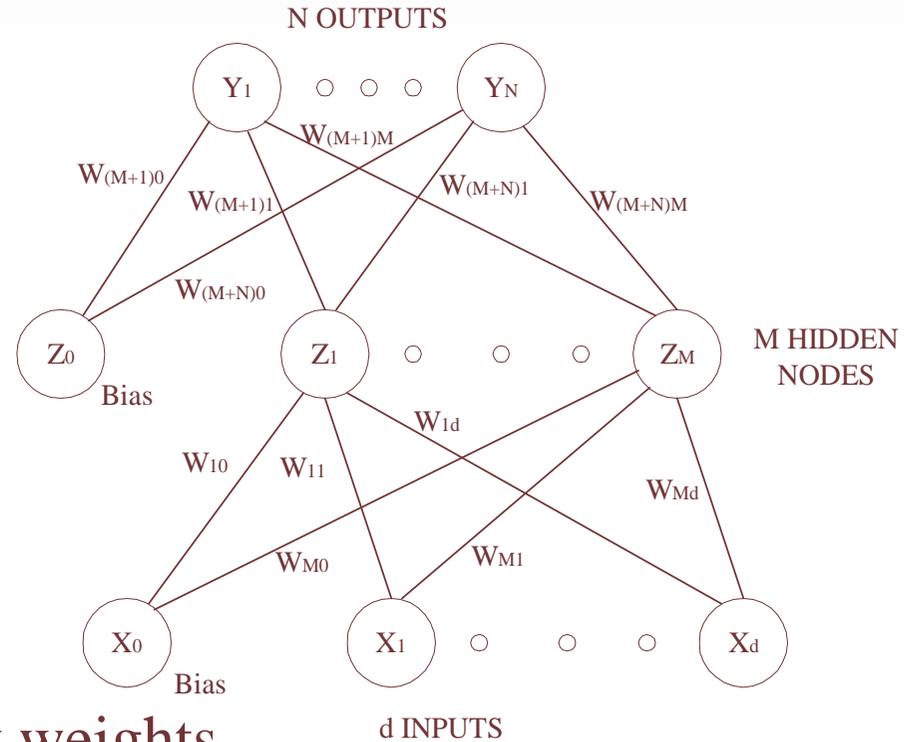
- Number of ordering heuristics available in literature.
- No heuristic guarantee that the end result of the conversion will be minimal.
- It is likely that one of heuristics available will convert the BDD, the problem is in finding it.
- Selection procedure developed using neural networks to try and achieve this.
- A set of ordering heuristics are used for selection.

# Ordering Heuristics – Neural Networks

- Neural networks provide a method to identify patterns.
- Pattern for conversion problem is between fault tree and ordering heuristic.
- The methodology involves:
  - A training phase, where patterns are learned.
  - A predictive phase, for unseen fault trees.

# Ordering Heuristics – Neural Networks

- Network consists of:
  - A layer of input nodes.
  - An output layer.
  - A hidden layer.



- Each layer connected by weights.
- Weights of connections are altered so as to move the result produced by the network toward the desired result.

# Methodology for Ordering Problem

- Fault tree characteristics (11) chosen intuitively.
- Characteristics are:
  - 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
  - Highest number of repeated events

# Methodology for Ordering Problem

- Ordering heuristics (6) available for selection:
  - Top-down, left-right.
  - Depth-first approach.
  - Priority depth-first approach.
  - Repeated event versions of each of the above.
- Form the outputs to neural network.
- Network trained using a large set of fault trees (from industry and random generation).
- Performance of network – predicted 14 out of 20 correct from set of test trees.

# Problems

- Ideal solution – minimal or near minimal BDD for all trees, to take full advantage of BDD methodology in commercial packages.
- Selection procedure shows potential for being able to generate minimal BDD for all trees.
- Areas for improvement – establishing which fault tree characteristics are required to establish pattern to desired ordering heuristic.
- Methodology to achieve this – JACOBIAN method.

## Using Jacobian Method – General Approach

- Back-propagation technique, similar to that used in training of neural network can be applied.
- Jacobian method – looks at the sensitivity of the outputs with respect to the inputs.
- In ordering problem, this is looking at sensitivity of ordering schemes with respect to fault tree characteristics.
- Those characteristics which strongly influence ordering methods will be highlighted.

## Using Jacobian Method – General Approach

- Sensitivity of output  $k$  to input  $i$  is defined as:

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

Each sensitivity is the derivative of the output with respect to the input.

## Using Jacobian Method – Procedure

- General procedure:
  - 1) Take a fault tree structure, analyse it for the desired characteristics, put through neural network selection procedure to find predicted outputs.
  - 2) Next the Jacobian for each output needs to be calculated.

$$J_{ki} = \underbrace{\sum_j w_{ji} z'_j}_{\text{Part 2}} \underbrace{\sum_l w_{lj} z'_k \delta_{kl}}_{\text{Part 1}}$$

## Using Jacobian Method – Procedure

- 2) The Jacobian values are calculated by recursively passing back through the network.

Initial summations 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 the connections between the hidden and input layer using part 2 of equation (3) are considered.

$$J_{ki} = \underbrace{\sum_j w_{ji} z'_j}_{\text{Part 2}} \underbrace{\sum_l w_{lj} z'_k \delta_{kl}}_{\text{Part 1}} \quad (3)$$

## Using Jacobian Method – Procedure

- 3) Steps 1 and 2 are repeated for all output nodes  $k$  with respect to all possible inputs  $i$ .

### Points to note:

- Each derivative is calculated with all other inputs held fixed.
- As the network represents a non-linear mapping, the elements of Jacobian matrix will not be constant, therefore the Jacobian must be re-evaluated for each new input (fault tree).

# Jacobian Method Application

- The Jacobian method has been applied to find the sensitivity of the ordering heuristics to the input fault tree characteristics.
- Jacobian values can be positive or negative, the magnitude is of interest, so modulus of Jacobian values used.
- The larger the Jacobian value the larger the influence.
- The sensitivity has been calculated using 198 input patterns.
- As the Jacobian must be re-evaluated for each new pattern, an average Jacobian value has been calculated.

# Jacobian Method Application

- The main concern is to determine whether the input characteristics are important.
- Need to look at – INPUTS.
- As a measure of this, the effect of input  $i$  has been averaged over all the outputs and over all the training patterns.
- The values produced are relatively small, yet there is a considerable difference in magnitude.

# Jacobian Method Application

- Rating the Jacobian values:

Input 0	43
Input 1	6
Input 2	9
Input 3	29
Input 4	39
Input 5	32
Input 6	2
Input 7	36
Input 8	100
Input 9	0
Input 10	16

- Largest = Input 8.  
(No. of gates with event inputs)
- The smallest = Input 9.  
(No. of gates with just gate inputs)
- Input 8 has twice influence of the second ranked Input.
- Almost 10 times the influence of Input 9.

# Jacobian Method Application

- Ranked characteristics:

Rank	Input Number	Characteristic	Jacobian average value
1	8	Number of gates with just events inputs	<b>0.047840</b>
2	0	Percentage of AND gates	<b>0.023896</b>
3	4	Number of levels	<b>0.022245</b>
4	7	Max number of gates	<b>0.020790</b>
5	5	Number of outputs from top gate	<b>0.019188</b>
6	3	Top gate type	<b>0.017626</b>
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

# Jacobian Method Application

- Interesting results were highlighted with regard to influential characteristics.
- Next step was to determine whether the more influential characteristics alone could still produce a neural network with the same or better predictive ability.
- A subset (of six) characteristics were taken to retrain the neural network.

## Re-Training – The Results

- Network architecture:
  - 6 input nodes.
  - 5 hidden nodes.
  - 6 output nodes.
- Network tested on same test set of fault tree structures.
- Predictive potential – 14 out of 20.
- Same predictive potential as when using eleven characteristics, suggesting 5 omitted characteristics provide no further information to help pattern recognition potential of network.

# Conclusions

- Using Jacobian provided insight into the important fault tree characteristics in the BDD conversion process.
- Reducing set of characteristics from 11 to 6 has yielded the same predictive potential when using the neural network technique.
- Thus, the workings of the neural network are now more efficient for the same response.
- Further work to enhance the predictive potential requires additional fault tree characteristics to be investigated. The Jacobian method can be used for this purpose.

# Conclusions

- Also the neural network selection mechanism can be improved by extending the range of ordering heuristics used to produce the BDD.
- This extension will involve further 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 the neural network approach.
- Ultimate aim to find necessary characteristics using this method to increase predictive potential to warrant using BDD approach in commercial package.