Comparison of Two New Approaches to Variable Ordering for Binary Decision Diagrams

Dr. Lisa Bartlett & Dr. John Andrews

Loughborough University

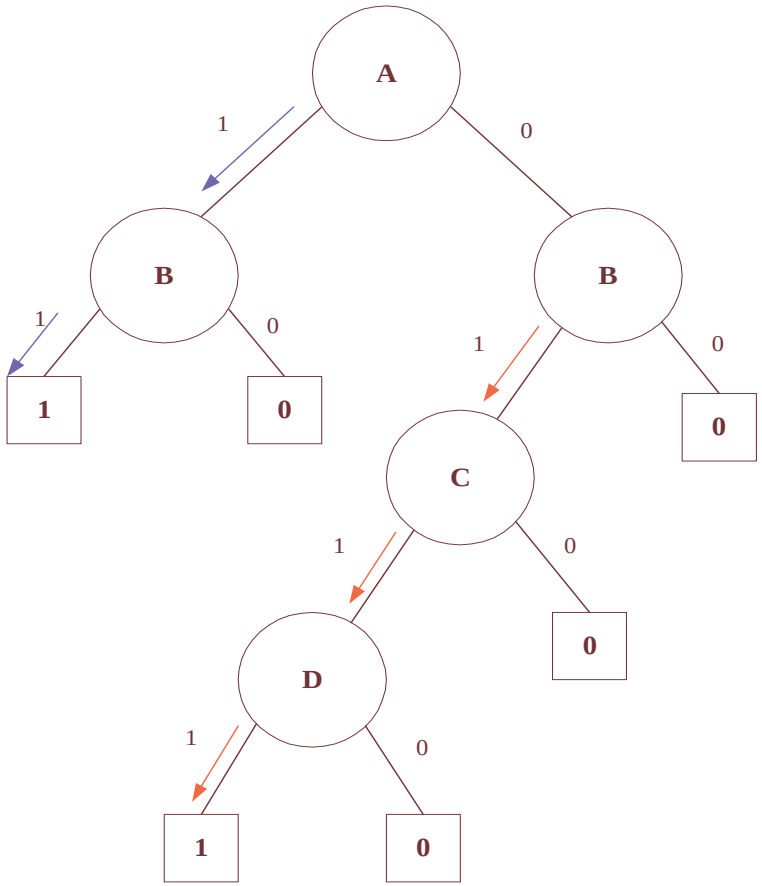
Summary of Presentation

- Binary Decision Diagram (BDD) Approach.
- Problems with BDD Approach -
  - Ordering the Basic Events (Variables).
- Solving Ordering Dilemma:
  - Neural Networks.
  - A New Approach - Structural Importance.

What is the BDD Approach?

- Problems in the Fault Tree Risk Assessment Technique.
  - BDD is an Improved Analysis Procedure.
- Fault tree is converted to the BDD representation.
  - Considers each components effect on the system.
- Conversion process requires the basic event variables to be placed in an order.
- Problem with ordering of variables.
  - Leads to non-minimal BDD structure.

The Binary Decision Diagram



- Fault Tree Converted to this diagrammatic representation.
- Directed Acyclic Graph.
- Paths start at root vertex.
- Terminate in either of two states.
- Cut sets lie on 1 branch to terminal 1 vertex.

Problems With BDD Approach

- Need to take an ordering of variables of fault tree to convert to BDD.
- The order variables are taken is critical to resulting size of BDD.
- Good ordering - efficient analysis.
  - Minimal BDD, minimal (near minimal) cut sets directly, optimal quantification process.
- Poor ordering - inefficient analysis / analysis not possible.
  - Non-minimal BDD, minimisation procedure required, quantification process longer.

Aim of Research

- To benefit from the increased efficiency and accuracy of the BDD technique the conversion process needs to be optimal for any fault tree.

- Currently a number of ordering procedures but all perform differently depending on the tree. Thus, the ordering problem needs to be solved.

- Two approaches -
  - Choosing the best ordering procedure from a set - neural networks.
  - Completely new procedure - structural importance.

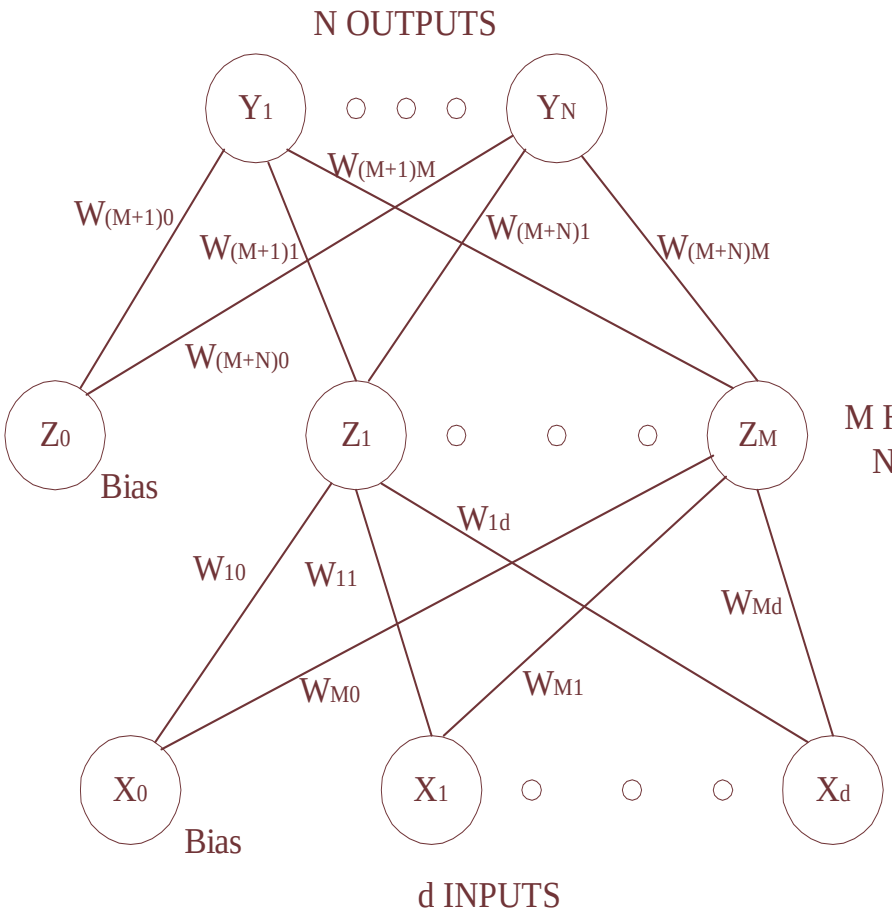
Approach 1 - Using Neural Networks

- Number of ordering heuristics in literature.
  - Top - down, left-right, repeated event procedures, weighting methods
- Size of BDD will vary according to tree.
- Can achieve good ordering with these heuristics, however need to know best one for chosen fault tree.
- Thus, need procedure to select best ordering heuristic from a set - map characteristics of fault tree to ordering best suited.

Why Neural Networks?

- Method of identifying patterns.
- Particular choice for a set of functions that map a set of input variables to a set of output variables.
- Multi-layer feedforward network has been applied.
- Two modes of operation: a training phase and a predictive phase.
- Training data set of fault trees with best known ordering heuristics. Take characteristics and train the network to recognise a new tree with same characteristics.

Diagrammatic Representation



- Series of layers.

- Connections running from every unit in one layer to every unit in the next layer.

- Connections are known as the *weights*.

- Control relationship between inputs and outputs.

- Hidden layer of non-linear functions.

Network for Ordering Problem

- Inputs - characteristics of fault tree - eleven chosen initially.
  - Percentage of AND gates, no. of repeated events, no. of basic events.
- Outputs - ordering heuristics - set of six selected.
- Training phase sets weights to model relationship between input characteristics and the best ordering procedure.
- Network can then be used for predictive purposes - choose ordering for previously unseen tree.

Results Using Neural Network

- Best Architecture: 11 - 5 - 6.
- Predicted 14 out of the 20 test trees with correct scheme preferences => 70 % chance of making correct choice for unseen tree.
- Remaining 30 percent of predictions - produce non-minimal BDD.
- The range of deviation from the minimal varied depending on the fault tree.

Problems with Current Heuristics

- Heuristics affected by how the fault tree was drawn.
- Many heuristics do not allow for components to be selected from different branches of the tree and lie next to each other in the ordering list.
- Dilemma of dealing with matched components in methods that assign weights to each basic event.

Desired Heuristic Properties

- Properties required in a good ordering heuristic seem to be:
  - The contribution of an event to the system failure mode must be reflected in the ordering produced.
  - The ordering must be robust i.e. the ordering must be dependent upon the logic function represented by the fault tree and not influenced by the way the fault tree has been drawn.
  - To uniquely map the fault tree onto a single event ordering.
- Structural importance measure was investigated.
  - ☒ This heuristic satisfies two out of the three points above.

Approach 2 - Structural Importance

- Want a method that considers the contribution an event makes.
- The importance measure signifies the role that the component plays in contributing /causing the occurrence of the top event.
- Uses structural importance measure of components.

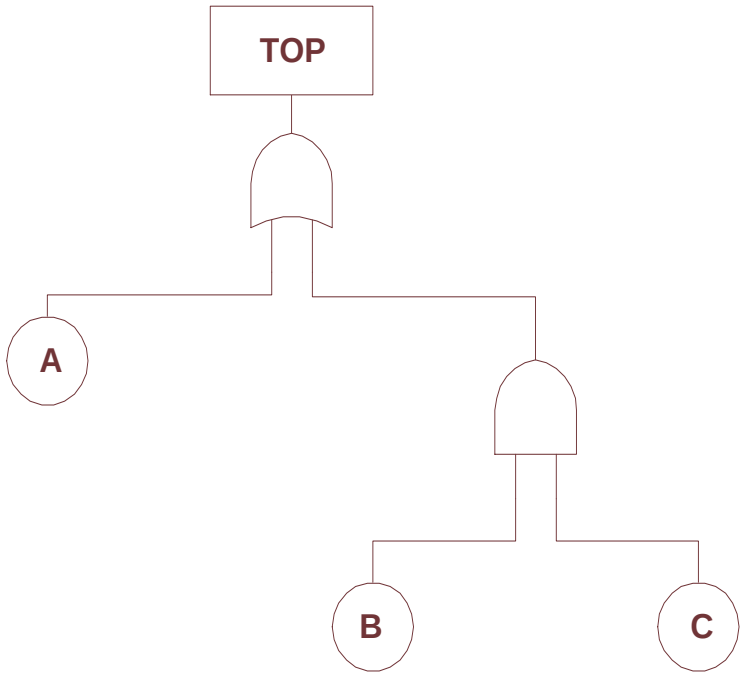
Importance Measures

- Deterministic Measure Used
  - assess the importance of a component to the system operation without considering the components probability of failure.

- IMP_j = $\frac{\textit{number of critical system states for component } j}{\textit{total number of states for the (n-1) remaining components}}$

- Critical State for Component, *j* -
  - is a state for the *n-1* remaining components such that a failure of component *j* causes the system to go from a working to a failed state.

Illustrating Structural Importance Measure



- Look at each component in turn.
- Look at all possible states for remaining components.
- BC, $\overline{B}C$, $B\overline{C}$, $\overline{B}\,\overline{C}$
- See if critical, i.e. is there a change in system state if component goes from working to failed state?

Mathematical Computation

- Birnbaum's Measure of criticality
  - $G_i(\mathbf{q}) = Q(1_i, \mathbf{q}) - Q(0_i, \mathbf{q})$
  - where $Q(\mathbf{q})$ is the probability that the system fails and
    - $Q(1_i, \mathbf{q}) = (q_1, q_2, \ldots..,q_{i-1}, 1, q_{i+1},\ldots\ldots q_n)$
    - $Q(0_i, \mathbf{q}) = (q_1, q_2, \ldots..,q_{i-1}, 1, q_{i+1},\ldots\ldots q_n)$.
  - with probability of remaining components equal to 1/2, gives -
    - $B_i = \{ Q(1_i, ½) - Q(0_i, ½) \}$
    - *The Structural Importance Measure*

How Generate Ordering?

- Top Event Logic Function: Top = AB + BDE
- Probability: $Q_{sys} = q_A q_B + q_B q_D q_E - q_A q_B q_D q_E$

- For component A:
- $(q_B + q_B q_D q_E - q_B q_D q_E) - (q_B q_D q_E) = 1/2 - 1/8 = 3/8$
- For component B:
- $(q_A + q_D q_E - q_A q_D q_E) - (0) = 5/8 - 0 = 5/8$
- For component D = E :
- $(q_A q_B + q_B q_E - q_A q_B q_E) - (q_A q_B) = 3/8 - 1/4 = 1/8$

- Thus, order => B < A < D < E

Results & Conclusions

- Tested against 225 fault trees.
- From six alternatives, compared against best result.
- Outperforms the best of a selection of six alternatives.
- Approximately 77% of all trees tested yielded a BDD of smaller or equal dimension.
- Difficulty in finding efficient algorithm to calculate measure directly from fault tree.

Conclusions

- Neural Network Approach -
  - Selection of heuristic from set of alternatives.
  - Allows variation depending on fault tree.
  - 70 % chance of producing minimal BDD.
- Structural Importance Approach -
  - Direct approach.
  - 77 % chance of producing BDD of equal or smaller dimension than previous best.
  - Efficient algorithm needs to be generated.