Advances in Reliability, Safety and Security

ESREL 2024
Monograph Book Series

Fault Tree Culling in the Dynamic and Dependent Tree Theory (D2T2) Framework through Approximation Techniques

Rundong Yan, John Andrews

Resilience Engineering Research Group, University of Nottingham, England

Organiser use only: Received date; revised date; accepted date

Abstract

This research explores an advanced approach to Fault Tree Analysis (FTA) by integrating Binary Decision Diagrams (BDDs) and addressing challenges associated with dependent failure events and variable ordering. Despite the advantages of BDDs, establishing an optimal variable order can be difficult, leading to computational inefficiencies or, in extreme cases, making BDD formulation impossible. To overcome these challenges, an approximation method is proposed within the Dynamic and Dependent Tree Theory (D^2T^2) framework, focusing on fault tree culling. The methodology strategically truncates minimal cut sets to yield the ones emphasising significant contributions to the top event. This enables the identification and prioritisation of critical failure events. The research further enhances the benefit of using D^2T^2 in addressing dependencies in failure events. This methodology holds significant promise in overcoming computational challenges in fault tree analysis, thereby ensuring the efficiency while ensuring the high accuracy of system reliability assessments. The implications extend to the advancement of safety and dependability in complex engineering systems.

Keywords: Fault tree analysis, Binary decision diagrams, Dependency models, Minimal cut set, Probability Approximation

1. Introduction

Fault Tree Analysis (FTA) stands as a fundamental tool in the assessment of system reliability and safety, providing a structured methodology for understanding and mitigating potential failure events. In 1970, Kinetic Tree Theory (KTT), introduced by Vesely, became a pivotal method for conducting FTA (Vesely, 1970). It consists of 2 stages. The first stage is to find the minimal cut sets (MCSs) - lists of basic events that are minimal, necessary, and sufficient to cause the top event. The second stage quantifies the probability or frequency of the system failure mode, i.e., the top event. However, a notable drawback of KTT lies in its assumption of independent occurrences of basic events, neglecting the impact of one or multiple basic failure events on the likelihood of others. Due to the occurrences of dependent failure events in systems reliability modelling, Dynamic and Dependent Tree Theory (D^2T^2), developed by Andrews and Tolo, emerged as a solution to overcoming some of the limitations of the traditional KTT (Andrews and Tolo, 2023).

The D^2T^2 framework aims to retain the fault tree structure for representing system failures, allowing transparency and compatibility with existing fault tree models. For complex systems, it requires the use of Petri nets or Markov models to efficiently analyse components with non-constant failure rates, dependencies, and complex maintenance processes. Then, the results obtained using Petri nets or Markov models are reintegrated into the FTA. Finally, the fault tree logic function can be converted to a Binary Decision Diagram (BDD) to calculate the system (Top event) failure probability and failure intensity. BDDs play a critical role in the field of reliability engineering, particularly in the analysis of fault trees. By utilising BDDs, the complex Boolean equations identified within fault tree representations can be transformed into a disjoint form. This transformation is able to facilitate the

exact quantification of system performance without the need to derive MCSs as intermediate results (Sinnamon and Andrews, 1997; Reay and Andrews, 2002; Remenyte-Prescott and Andrews, 2008; D. R. Prescott et al., 2009).

While the method makes good use of BDDs for their benefits, it also faces some challenges (Andrews and Tolo, 2023). In particular, determining the optimal variable ordering for fault tree solutions with BDDs can be intricate in certain scenarios, leading to additional computational complexities (Ibáñez-Llano et al., 2010; Rivero Oliva et al., 2018). In extreme cases, the creation of the BDD may become excessively challenging. Consequently, using an approximation becomes a practical and ideal way to conduct the analysis. This paper introduces a novel fault tree culling methodology integrated into the D²T² framework, with the primary objective of identifying the most significant MCSs relevant to the top event. Subsequently, the critical MCSs or the reduced fault tree, constructed by these important MCSs, undergo a quantitative analysis employing a BDD, as elaborated within the D²T² framework.

The subsequent sections of this paper are structured as follows: Section 2 provides a brief review of the D²T² methodology. Section 3 describes the culling method in detail. Section 4 introduces a case study designed to demonstrate the proposed methodology, while Section 5 presents the application of the method to the aforementioned case study. Finally, Section 6 concludes the paper by summarising key findings and outlines potential future research.

2. Review of Dynamic and Dependent Tree Theory (D²T²)

Dynamic and Dependent Tree Theory (D²T²) extends the capabilities of Kinetic Tree Theory (KTT) for advanced reliability analysis. The theory is developed to analyse complex fault trees characterised by dependencies, sequences among basic events, and non-constant failure or repair rates. The theory utilises a combination of Binary Decision Diagrams (BDDs), Petri Nets (PN), and Markov methods (Rauzy, Gauthier, and Leduc, 2007; Yevkin, 2016; Rundong Yan, Dunnett, and Andrews, 2023). The algorithm executing D²T² can be divided into 7 steps.

1. Identify the initiators and enablers.
2. Identify the dependency groups existing in the system. It should be noted that these dependency groups are independent of all other events.
3. Compute the probabilities or frequencies of the independent events using the traditional component failure models based on the raw failure data such as failure rates.
4. Reorganise the fault tree structure into independent sub-modules, which can be solved effectively. This step is so-called modularization. This involves a two-stage approach. Initially, the Factor approach's three processes, i.e. contraction, factorisation, and extraction, are repeatedly applied to maximumly reduce the fault tree structure (Reay and Andrews, 2002). Then, the linear time algorithm of Dutuit and Rauzy can be conducted to identify independent gates in the fault tree structure (Dutuit and Rauzy, 1996).
5. Develop the models using the PN modelling method and/or Markov modelling method to quantitatively assess the dependency groups.
6. Construct the BDD for the modularised fault tree.
7. Compute the top event probability and intensity result by integrating the PN and Markov models' outcomes into the BDD.

Petri nets (PNs) have been widely adopted in the modelling of various complex systems (D. Prescott and Andrews, 2013; Davies and Andrews, 2021; R. Yan, Dunnett, and Jackson, 2022). In the D²T² framework, PNs are adopted to model complex event dependencies and complex maintenance processes. It provides a direct bipartite graphical representation of a system, enabling the analysis and simulation of system behaviour. They consist of two types of elements. The first one known as places (shown as circles in Fig. 1), indicates the states of the system or component. Transitions (shown as squares) represent the actions or events that can change the system's state. The arrows or edges, known as arcs in Petri nets, link places and transitions.

A transition can be activated when the number of tokens within each input place is greater than or equal to the respective weights assigned to the arcs inputting to the transition. In Fig. 1, the transition is fired after time *t*. One token is removed from the input place and one token is produced in the output place. The marking of tokens within a PN model represents the state of the system modelled.

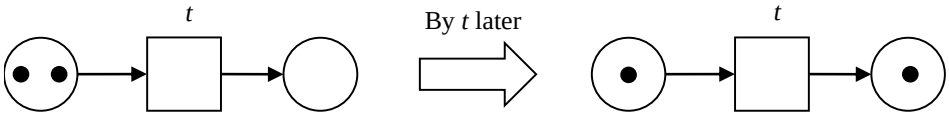

Fig. 1. An example Petri net

Markov models are adopted in the D^2T^2 framework to characterise systems with dependencies, where the constancy of failure and repair rates is a key assumption. The advantage lies in Markov models' capability to represent the stochastic behaviour of systems that undergo discrete variations over time (Chiacchio et al., 2011; Yevkin, 2016). Markov models which vary discretely in time are known as Markov chains. They consist of two different elements: states (nodes) and transitions (edges). Each node represents a specific condition or configuration of the system at a given moment. Each connecting one state to another is assigned a probability. A simple two-state Markov Chain with states 'W' and 'F' is given in Fig. 2. It represents a single repairable component characterised by a failure rate of λ and a repair rate of ν.

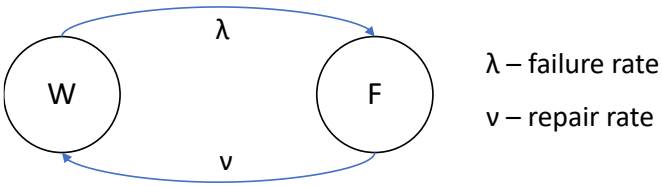

Fig. 2. An example Markov model

3. Fault Tree Culling Method for D^2T^2

Although D^2T^2 has made a great effort to minimise the size of the BDD and the number of path calculations required to be performed via modularization to ensure the algorithm is efficient, certain inherent constraints persist within the methodology in some circumstances (Tolo and Andrews, 2023; Andrews and Tolo, 2023). For example, sometimes, solving the fault tree of a complex system using BDDs with a good ordering of the variables cannot be formulated. In such scenarios, adopting an approximation method is essential to derive approximate values for the top event probability and intensity close to their exact counterparts. This approximation method should be conducted prior to the construction of the BDD, i.e. before Step 6 of D^2T^2 detailed in Section 2. The steps for implementing the appropriation method developed in the paper as a supplementary of D^2T^2 to counter these special scenarios are given in the following.

1. Compute the MCSs of the modularized fault tree using Boolean algebra (Rivero Oliva et al., 2018).
2. Compute the probability and frequency of each MCS.
3. Cull the MCSs by producing only the most significant MCSs relevant to the top event. There are mainly three different methods for culling the MCSs (Sinnamon and Andrews, 1996).
    i. Probability Culling: Keep MCSs with a probability greater than or equal to a specified cutoff value.
    ii. Frequency Culling: Keep MCSs with a frequency/failure intensity greater than or equal to a specified cutoff value.
    iii. Order Culling: Produce only those MCSs with an order less than or equal to the specified threshold. In other words, retain only those MCSs consisting of an equivalent or fewer number of basic events compared to the maximum allowable number of basic events in MCSs.

The probability culling and frequency culling methods are applicable when assessing the probability or intensity of the top event, respectively. The BDD can then be developed based on the remaining important MCSs. Finally, the determination of the approximate top event probability or intensity is achieved by solving the BDD.

4. Pressure Vessel Cooling System Case Study

The case study analysed by (Andrews and Tolo, 2023) to showcase the application of D^2T^2, which involves a pressure vessel cooling system, is revisited here to demonstrate the utilisation of the culling method for approximating the top event probability. The schematic of the cooling system is illustrated in Fig. 3.

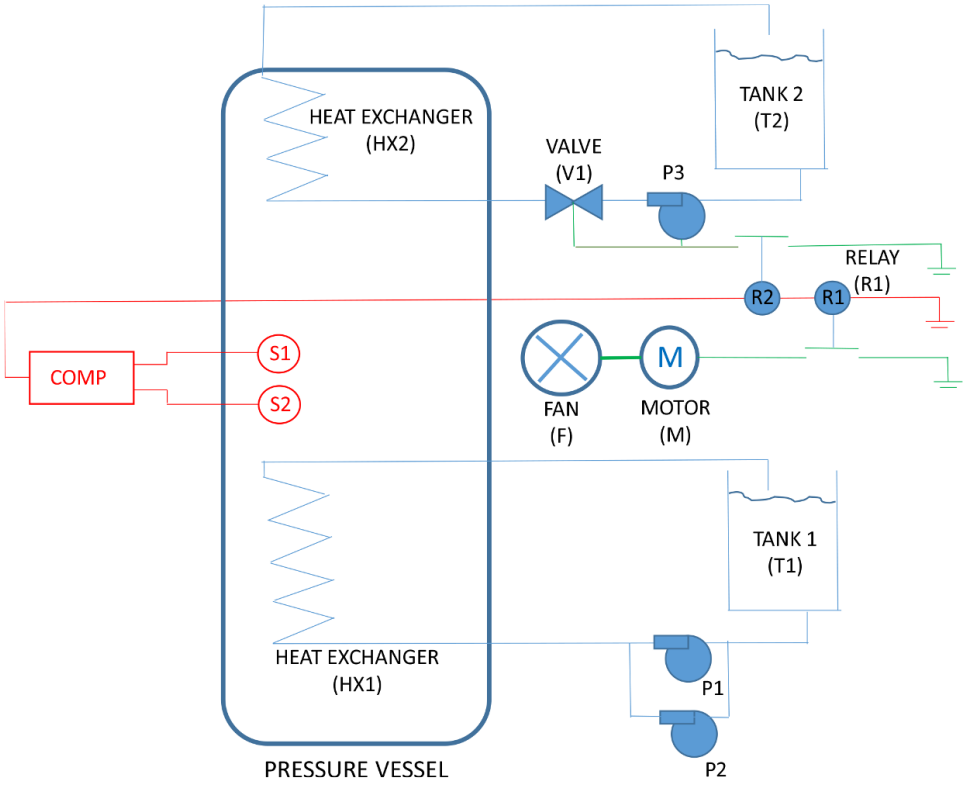

Fig. 3. Pressure vessel cooling system case study (Andrews and Tolo, 2023).

The system is designed to provide cooling for an exothermic chemical reaction. As shown in the lower part of Fig. 3, the primary cooling system involves the cycling of water from tank T1 to heat exchanger Hx1 through two pumps, P1 and P2, both of which are actuated by a shared power source denoted as PoW. In the event of a malfunction in the primary cooling system, the vessel temperature monitored by the thermocouples S1 and S2 will increase. Once the temperature reaching a set threshold is detected by either of the thermocouples, the computer (Comp) initiates a sequence wherein relays R1 and R2 are de-energised, activating two alternative cooling systems. The first system incorporates water supply T2, heat exchanger Hx2, and a singular pump, denoted as P3. The second cooling mechanism involves a fan, designated as F, which is driven by a motor denoted as M. In the event of a primary cooling system failure, both the auxiliary systems are activated, and they must operate continuously for an extended duration of 30 days.

5. Case Study Analysis

To analyse the reliability of the system using FTA, 'Pressure Vessel Cooling Fails' is defined as the top event. The fault tree for the top event developed by (Andrews and Tolo, 2023) is provided in Fig. 4. In addition, there are three dependency groups in the system, which are {*P1*, *P2*}, {*Hx1*, *Hx2*}, and {*P3S*, *P3R*}, respectively. The probability of each basic event in the fault tree can be found in (Andrews and Tolo, 2023).

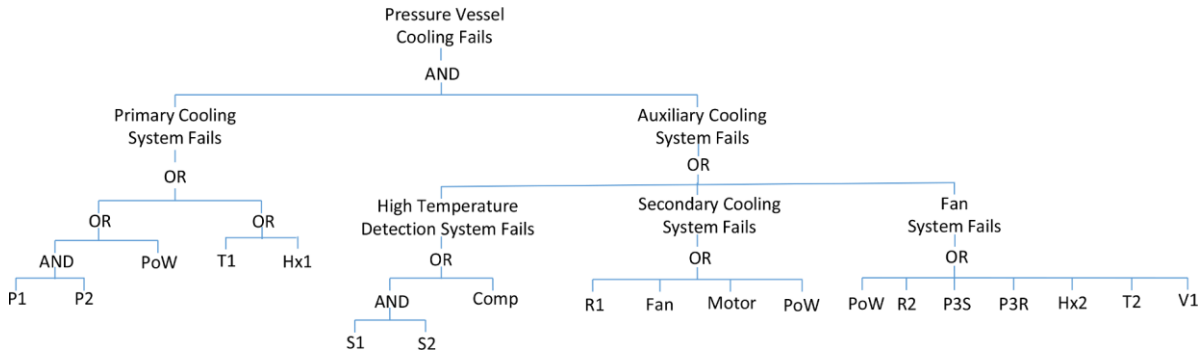

Fig. 4. Pressure vessel cooling system fails fault tree (Andrews and Tolo, 2023).

By conducting Steps 1 to 5 of D^2T^2, the probability and intensity of each basic event can be obtained. The solutions for the complex dependency groups identified in the system are calculated with appropriate PN or Markov models. Then the approximation method can be implemented.

Firstly, the MCSs of the fault tree developed for the top event 'Pressure Vessel Cooling Fails' are derived using Boolean algebra. For the fault tree, three key laws of Boolean algebra are used (Mahmud, 2019; Banov, Šimić, and Grgić, 2020; Ye et al., 2021). It is worth mentioning that more laws should be adopted if the fault tree being analysed is noncoherent. In addition, it should be noted that the symbols '+' (plus) and '.' (dot) used in the following Boolean equations represent the logical OR and logical AND operations, respectively.

1. Distributive Law: Describes the relationship between logical AND and logical OR operations.

$$A.(B+C) = A.B + A.C \tag{1}$$

2. Idempotent Law: States that combining a Boolean variable with itself using either logical OR or logical AND yields the variable itself. This helps to remove repeated cut sets and failure events.

   i. $A.A = A$ (2)

   ii. $A + A = A$ (3)

3. Absorption Law: States that the logical OR of a variable with the logical AND of that variable and another variable is equivalent to the variable itself.

$$A + (A.B) = A \tag{4}$$

The top event can be expressed in terms of its inputs using Boolean algebra. Starting at the top of the tree, the top event can be expressed as Equation 5.

$$TOP = (Primary\ Cooling\ System\ Fails).(Auxiliary\ Cooling\ System\ Fails) \tag{5}$$

Then, Substitute the expression for each gate working down through the tree as shown in Equation 6 and expand and simplify the expression using the laws of Boolean algebra, if possible.

$$TOP = (P1.P2 + PoW + T1 + Hx1).(S1.S2 + Comp + R1 + Fan + Motor + \cdots) \tag{6}$$

Ultimately, the expression can be reduced to a minimal form given in Equation 7. It is also known as the disjunctive normal form or minimal sum-of-products form. The MCSs are the terms between the plus signs.

$$TOP = P1.P2.S1.S2 + P1.P2.Comp + P1.P2.R1 + P1.P2.Fan + P1.P2.Motor + +P1.P2.R2 \ldots\ldots \tag{7}$$

In total, 34 MCSs are identified for the fault tree. In order to approximate the top event probability, the probability of each MCS is calculated and then ranked as shown in Table 1.

Table 1. The minimal cut set probability.

Minimal Cut Sets	Probability	Rank
PoW	1.000E-03	1
P1.P2.V1	9.618E-04	2
P1.P2.Comp	9.404E-05	3
P1.P2.P3	9.037E-05	4
P1.P2.Hx2	2.313E-05	5
P1.P2.R1	1.915E-05	6
⋮	⋮	⋮

By setting a minimum probability threshold for keeping the important MCSs to 9.000E-05, four MCSs, i.e. {PoW}, {P1.P2.V1}, {P1.P2.Comp} , and {P1.P2.P3}, are retained. Based on these important MCSs, a BDD can be constructed as described in Step 6 of D^2T^2 as shown in Fig. 5.

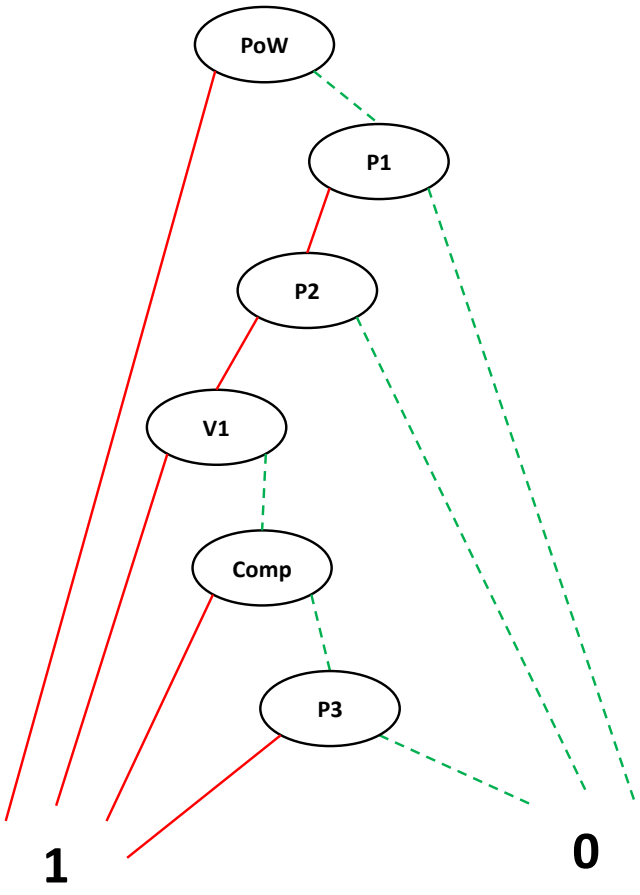

Fig. 5. BDD structure resulting from the conversion of the important MCSs.

The individual decision nodes (ellipses) in the BDD correspond to Boolean variables, each representing a basic event. Each decision node has two outgoing edges. One edge (red solid lines in the Figure) represents the True (or 1) value of the value, which means the basic event occurs. Another edge (green dotted lines in the Figure) represents the False (or 0) value of the value, which means the basic event does not occur. At the bottom of the Figure, there are two terminal modes representing the final output of the Boolean function. They are labelled with 1 and 0, meaning the occurrence or non-occurrence of the top event, respectively.

In the BDD, there are four disjoint failure paths. The events contributing to each path are identified and then the probability of each path is calculated, shown in Table 2.

Table 2. Failure paths for the BDD.

Path No.	Events Contribution	Probability
1	$q(PoW)$	1.000E-03
2	$q(\overline{PoW}) \times q(P1,P2) \times q(V1)$	9.608E-04
3	$q(\overline{PoW}) \times q(P1,P2) \times q(\overline{V1}) \times q(Comp)$	4.110E-05
4	$q(\overline{PoW}) \times q(P1,P2) \times q(\overline{V1}) \times q(\overline{Comp}) \times q(P3)$	3.732E-05

Finally, the approximated probability of the overall system failure or top event can be calculated by summing the probability of each disjoint failure path (Tolo and Andrews, 2023). This summation is expressed as Equation 8.

$$Q_{system} = \sum_{i=1}^{4} q(P_i) = 0.002039272 \tag{8}$$

In comparing the value obtained with the exact top event probability of 0.0020906577 obtained in (Andrews and Tolo, 2023), a difference of 2.46% is observed. It is found that the approximated probability obtained via the fault tree culling method is lower than the exact true probability. This discrepancy suggests that the method will lead to an underestimation of the real value.

This underestimation is attributed to the selective retention of only the most significant MCSs, while disregarding other combinations of basic events that may contribute to the occurrence of the top event. Consequently, the neglect of these basic event combinations introduces a bias toward a diminished probability estimate, thereby compromising the accuracy of the method. Nevertheless, despite the observed underestimation, the culling method still offers a reasonably robust estimation of the problem. By increasing the number of MCSs kept after culling, the method can achieve a more accurate value.

6. Conclusion

In conclusion, this research introduces a novel fault tree culling method by truncating MCSs within the D^2T^2 framework. This approach emphasises the identification and prioritisation of critical event combinations, leading to a more focused analysis that highlights significant contributions to the top event of fault trees. By doing this, the size and complexity of the BDD required to be constructed using D^2T^2 can be reduced to an acceptable level. It should be highlighted that the approximation method can lead to an underestimation of the top event probability. However, despite this limitation, the method still yields a reasonable approximation of the problem. Therefore, the flexibility and adaptability of the method are able to empower users to enhance the accuracy of their assessments by adjusting the parameters to achieve an optimal balance between computational efficiency and a more precise approximation of failure probabilities.

Acknowledgements

This work was supported by the Lloyd's Register Foundation, a charitable foundation in the U.K. helping to protect life and property by supporting engineering-related education, public engagement, and the application of research.

References

Andrews, J., and S. Tolo. 2023. Dynamic and Dependent Tree Theory (D2T2): A Framework for the Analysis of Fault Trees with Dependent Basic Events. *Reliability Engineering and System Safety 230* (*May 2022*): 108959.

Banov, R., Z. Šimić, and D. Grgić. 2020. A New Heuristics for the Event Ordering in Binary Decision Diagram Applied in Fault Tree Analysis. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability 234* (*2*): 397–406.

Chiacchio, F., L. Compagno, D.D. Urso, G. Manno, and N. Trapani. 2011. Dynamic Fault Trees Resolution : A Conscious Trade-off between Analytical and Simulative Approaches. *Reliability Engineering and System Safety 96* (*11*): 1515–26.

Davies, B., and J. Andrews. 2021. The Impact of Summer Heatwaves on Railway Track Geometry Maintenance. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit 235* (*9*): 1158–71.

Dutuit, Y., and A. Rauzy. 1996. A Linear-Time Algorithm to Find Modules of Fault Trees. *IEEE Transactions on Reliability 45* (*3*): 422–25.

Ibáñez-Llano, C., A. Rauzy, E. Meléndez, and F. Nieto. 2010. Hybrid Approach for the Assessment of PSA Models by Means of Binary Decision Diagrams. *Reliability Engineering and System Safety 95* (*10*): 1076–92.

Mahmud, N. 2019. A Minimization Algorithm for Automata Generated Fault Trees with Priority Gates. *Software Quality Journal 27* (*3*): 1015–43.

Prescott, D., and J. Andrews. 2013. A Track Ballast Maintenance and Inspection Model for a Rail Network. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability 227* (*3*): 251–66.

Prescott, D.R., R. Remenyte-Prescott, S. Reed, J.D. Andrews, and C.G. Downes. 2009. A Reliability Analysis Method Using Binary Decision Diagrams in Phased Mission Planning. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability 223* (*2*): 133–43.

Rauzy, A.B., J. Gauthier, and X. Leduc. 2007. Assessment of Large Automatically Generated Fault Trees by Means of Binary Decision Diagrams. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability 221* (*2*): 95–105.

Reay, K.A., and J.D. Andrews. 2002. A Fault Tree Analysis Strategy Using Binary Decision Diagrams. *Reliability Engineering and System Safety 78* (*1*): 45–56.

Remenyte-Prescott, R., and J.D. Andrews. 2008. An Enhanced Component Connection Method for Conversion of Fault Trees to Binary Decision Diagrams. *Reliability Engineering & System Safety 93* (*10*): 1543–50.

Rivero Oliva, J. de J., J. Salomón Llanes, M. Perdomo Ojeda, and A. Torres Valle. 2018. Advanced Combinatorial Method for Solving Complex Fault Trees. *Annals of Nuclear Energy 120*: 666–81.

Sinnamon, R.M., and J.D. Andrews. 1996. Fault Tree Analysis and Binary Decision Diagrams. In *Proceedings of 1996 Annual Reliability and Maintainability Symposium*, 215–22. IEEE.

Sinnamon, R.M., and J.D. Andrews. 1997. Improved Accuracy in Quantitative Fault Tree Analysis. *Quality and Reliability Engineering International 13* (*5*): 285–92.

Tolo, S., and J. Andrews. 2023. Fault Tree Analysis Including Component Dependencies. *IEEE Transactions on Reliability PP* (*2*): 1–9.

Vesely, W.E. 1970. A Time-Dependent Methodology for Fault Tree Evaluation. *Nuclear Engineering and Design 13* (*2*): 337–60.

Yan, R., S.J. Dunnett, and L.M. Jackson. 2022. Model-Based Research for Aiding Decision-Making During the Design and Operation of Multi-Load Automated Guided Vehicle Systems. *Reliability Engineering & System Safety 219* (March): 108264.

Yan, Rundong, S. Dunnett, and J. Andrews. 2023. A Petri Net Model-Based Resilience Analysis of Nuclear Power Plants under the Threat of Natural Hazards. *Reliability Engineering & System Safety 230* (February): 108979.

Ye, L., E. Li, D. Zhao, S. Xiong, S. Zhou, and J. Xiang. 2021. An Efficient Approximation for Quantitative Analysis of Dynamic Fault Trees. In , 242–52. IEEE.

Yevkin, O. 2016. An Efficient Approximate Markov Chain Method in Dynamic Fault Tree Analysis. *Quality and Reliability Engineering International 32* (*4*): 1509–20.