Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 58-4 (2024) 378–383

A Deductive Fault Analysis Method Based on Hypergraphs

Carlo Vitucci [*] **Thomas Westerbäck** [**] **Daniel Sundmark** [***]
Håkan Forsberg [****] **Thomas Nolte** [†] **Marcus Jägemar** [‡]

[*] *Ericsson AB, Stockholm, Sweden (e-mail: carlo.vitucci@ericsson.com).*
[**] *Mälardalen University, Västerås, Sweden (e-mail: thomas.westerback@mdu.se)*
[***] *Mälardalen University, Västerås, Sweden (e-mail: daniel.sundmark@mdu.se)*
[****] *Mälardalen University, Västerås, Sweden (e-mail: hakan.forsberg@mdu.se)*
[†] *Mälardalen University, Västerås, Sweden (e-mail: thomas.nolte@mdu.se)*
[‡] *Ericsson AB, Stockholm, Sweden (e-mail: marcus.jagemar@ericsson.com)*

Abstract: Fault tree analysis is a system malfunction hazard evaluation quantitative and qualitative procedure. The method is well-known and widely used, especially in the safety systems domain, where it is a mandatory integral part of the so-called "Hazard Evaluation" documentation. This paper proposes an alternative or complementary deductive fault analysis method: it uses system topology to build a hypergraph representation of the system to identify component criticality and support loss of functionality probability evaluation. Once automated, the proposed method seems promising when the system engineers explore the different architectures. They may have indication about architecture's reliability without continuous feedback from the system safety team. The system safety team must check the solution once the engineers select the final architecture. They can also use the proposed method to validate the correctness of the fault tree analysis.

Keywords: fault analysis, hypergraph, loss of function probability, cut set level

1. INTRODUCTION

"Hazard Evaluation Procedures" are the quantitative and qualitative system analysis techniques used to identify possible system failure conditions and their failure rate Some of the most commonly used techniques are Fault Tree Analysis (FTA), Failure Mode and Effects Analysis (FMEA), and Root Cause Analysis (RCA) (Huang et al. (2020)). The quantitative procedures received an "attention reboot" from the fault prediction research, that is, the analysis of the working condition based on the probabilistic occurrence of event or failure condition to plan a preventive set of actions in advance.

1.1 Fault Tree Analysis

H.A. Watson of American Bell Telephone graduate school introduced the fault tree analysis in 1961 for reliability analysis of a missile shoot control system (Ericson (1999)). FTA is a quantitative and qualitative methodology that aims to identify component events relationship and today is a mandatory analysis method for safety system engineering (Ruijters and Stoelinga (2015)). It is commonly used by various industries during the reliability and safety analysis of complex system (Bhattacharyya and Cheliyan (2019)). Starting from a complete and general system fault analysis, FTA, in a top-down approach, considers the fault impact of the failure in a single component to the system malfunction. The top-down analysis is a deductive method. Eight steps summarize the reliability analysis process (EuroControl (2001)):

a) **FTA goal definition**: define the boundaries of interest.
b) **Top level identification**: specify the problem to be analyzed.
c) **FTA resolution**: define conditions that lead to the problem.
d) **FTA Design**: solve the fault tree for the combination of events.
e) **FTA evaluation**: perform quantitative analysis to evaluate system performance.
f) **Outcomes interpretation**: find potential hazard and place appropriate measure.
g) **FTA connection revised**: Explore each branch in successive levels of detail.
h) **FTA scope revised**: identify possible failure dependency and adjust the model.

From a top-level loss of function failure condition, the method involves building a tree graph as a logical representation of events and their connections. Events are possible failure conditions with their failure occurrence, where a lower event produces higher events (connection). The tree elements are logic gates (and, or, nor, xor, nand, etc.) representing the logical correlation between lower events and higher events. FTA allows the highest event probability evaluation for bi-stable behavior components (either is working or is not working), which is a direct consequence of having logic gates as tree elements. FTA is also a preventive method because it helps to identify which (faulty) component contributes most to a system failure and its occurrence, and therefore understand for which components it is more appropriate to implement fault tolerance mechanisms to minimize the probability of failure for the system. There are two types of fault analysis: static and dynamic. Dynamic Fault Tree Analysis (DFTA) is an extension of the Static Fault Tree Analysis (SFTA) as a response to the poor ability of SFTA to take into account dynamic aspects of fault tolerance (Dugan et al. (1992); Vesely et al. (2002)), such as, for example, the presence of redundancy mechanisms (Aslansefat et al. (2020)). FTA is a very effective risk assessment method but has some challenges:

- FTA doesn't capture any time-related or delay factors (Ericson (2005)).

- FTA for complex and large systems becomes of considerable size, time-consuming, and prone to design mistakes (Sinnamon and Andrews (1997); Khakzad et al. (2013)).

- It is not easy to detect event duplication in multiple diagram path errors. Common cause failures are not always obvious (Kabir (2017)).

- FTA is a helpful support for root cause diagnosis (Ruijters and Stoelinga (2015)), particularly in highlighting crucial insights, despite its limitation of analysing the top event only (Kritzinger (2017)).

- FTA needs skilled individuals to understand the logic gates, at least when more complicated gates such as inhibit, house or conditional events are used (Ferdous et al. (2007)).

2. RELATED WORK

Most researchers agreed on the FTA challenges listed at the end of Section 1, but, as per our knowledge, only a few studies explored alternative methods. Aslansefat et al. (2020)) and Zhu and Zhang (2022)) provided a significant fault analysis research survey. It is possible to identify three areas of interest for the researchers:

Improving top-level loss functionality probability calculation. The main goal for the researchers in this group is probability evaluation. Wu et al. (2007) use the fuzzy theory to calculate the probability of fault estimation. Continuous-time Markov chain (CTMC) was the forerunner of stochastic dynamic modelling. Chen et al. (2013) describe a method using a temporal fault tree and Markov chain. Huang and Chang (2007) propose a hierarchical approach by modularising the fault tree. Zhu et al. (2014) use stochastic logic as templates for static and dynamic gates. *However, by solely focusing on probability calculation, they overlook the critical aspects listed at the end of section 1.1.*

$\backslash E$ $V\backslash$	e_1	e_2	e_3	e_4	e_5	e_6
v_1	1	1	0	1	1	1
v_2	1	0	1	1	0	1
v_3	0	1	1	0	0	1
v_4	0	0	1	1	0	1
v_5	0	0	0	1	0	1

Table 1. Hypergraph Incidence Matrix

Compute-based FTA algorithms. A method to cope with the FTA complexity for large systems is to design an algorithm introducing compute-based FTA calculation (Wang et al. (2012)) Compute-aid methods moves the complexity of the FTA design in the complexity of the algorithms. Rao et al. (2009); Yevkin (2010); Dai et al. (2011) propose a solution process for each dynamic gate using Monte Carlo simulation. Raptodimos and Lazakis (2017) propose combining DFT with machine learning. Nguyen et al. (2015) introduces a new method based on the Pauli network. *However they don't represent an actual alternative to FTA usage, as our proposed method could be. Moreover, by removing any dependency to the logic gate, our proposed method could simplify the algorithm for a compute-aid fault analysis.*

FTA alternative. Lakehal et al. (2019) uses the Bayesian network method that is suitable for the fault analysis of non-bi-stable system components. Roth et al. (2015) introduced the matrix-based model to support fault trees with a huge number of elements. Kabir et al. use the Pandora technique to extend FTA for dynamic systems. They described translating the Pandora-based fault tree to Bayesian Networks (Kabir et al. (2014)) Bouissou and Bon (2003) introduced the Boolean logic-driven Markov processes (BDMP), a method to model complex dynamic systems. Eventually, it is worth mentioning two tools providing support fault tree analysis: CORAL (Boudali et al. (2010)), and DFTCalc (Arnold et al. (2013); Guck et al. (2015)) providing efficient fault tree modelling via compact representations. *All methods claim to be an alternative to the traditional FTA method in facilitating its automation. Still, they do so at the expense of high implementation complexity and practicality in use.*

3. CONCEPTS AND DEFINITIONS

The problem on Seven Bridges of Königsberg by L. Euler is regarded by many to be the first published work on graph theory (Biggs et al. (1998)). Graphs are a mathematical concept used to model pairwise relations between objects. In this context ordinary graphs are made up by vertices and edges where an edge is connected to two vertices. Hypergraphs were introduced by C. Berge in the 1960s (Berge (1984)) as a generalization of graphs where edges join any number of vertices.

***Definition** 1.* A (finite) *Hypergraph* is a pair $\mathcal{H} = (V, E)$ where V is a finite set of elements called *vertices*, E is a finite set of elements called *edges*, and $e \subseteq V$ for each element $e \in E$.

Hypergraphs can be represented using incidence matrices.

***Definition** 2.* The *incidence matrix* $B = (b_{ij})_{m \times n}$ of a hypergraph $\mathcal{H} = (V, E)$ with $V = \{v_1, \ldots, v_m\}$ and

$E = \{e_1, \ldots, e_n\}$ is defined by

$$b_{ij} = \begin{cases} 1 \text{ if } v_i \in e_j, \\ 0 \text{ otherwise.} \end{cases}$$

***Example** 1.* Let $\mathcal{H} = (V, E)$ be the hypergraph where

$$V = \{v_1, v_2, v_3, v_4, v_5\},\ E = \{e_1, e_2, e_3, e_4, e_5, e_6\},$$

$$e_1 = \{v_1, v_2\},\quad e_2 = \{v_1, v_3\},\ e_3 = \{v_2, v_3, v_4\},$$
$$e_4 = \{v_1, v_2, v_4, v_5\},\ e_5 = \{v_1\},\quad e_6 = \{v_1, v_2, v_3, v_4, v_5\}.$$

Then Table 1 is the incidence matrix of $\mathcal{H}$.

4. HYPERGRAPH DEDUCTIVE METHOD

This section describes the proposed deductive fault analysis procedure called Hypergraph Analysis Method (HAM). 6 steps describe the introduced process

a) **HAM goal definition**: define the boundaries of interest. Similarly to what happens with the FTA process, even in the proposed method, the first step is defining the boundaries of interest, that is, the selection of the system functionality for which to calculate the loss of a selected function probability.

b) **Independent Failure chain identification**: The second step of the HAM process is the analysis of the system architecture or electrical diagram to identify the "failure path". A failure path is an independent system components subset that, in a bi-stable behavior condition, can compromise the functionality under investigation if it is not working. A failure chain is a combination of components within a failure path that cause a failure condition for the path.

c) **HAM resolution tuning**: it is always possible to see complex and large systems as the sum of multiple subsystems or events. Any subsystem can then be considered an independent system, each with fault paths and chains. HAM can be applied hierarchically, building a bigger Hypergraph from the subsystem Hypergraph based on the deducted relationship between subsystems in the fault path. Fig. 1 is a graphical example. Note how any subsystem in a complex system is also one of its fault paths.

d) **Hypergraph incidence matrix identification**: The components (or events) identified in the second step represent the vertices, and the failure chain represents the edges of the hypergraph $\mathcal{H}$ for the HAM method. By using

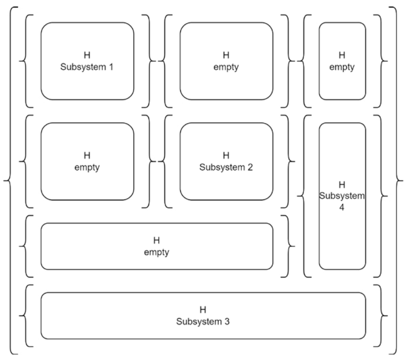


Fig. 1. Example of hypergraph incidence matrix for complex and large system

vertices as rows and edges as columns, it is possible to build the incidence matrix of $\mathcal{H}$, based on Definition 2.

e) **Hypergraph coherency and validity check**: Duplication of e_i will be evident, and a simple incidence matrix inspection allows the removal of duplication from the matrix. *In that sense, the HAM helps to detect error path duplication, recognized as a drawback for the FTA.*

f) **Outcome calculation and evaluation**: The system *incidence matrix* can be used to evaluate the loss of function and the system fault tolerance parameters.

***Definition** 3.* In fault analysis domain, a **Cut Set** is a set of basic events whose simultaneous occurrence ensures that the TOP event occurs. A **Minimal Cut Set** is a cut set that cannot be reduced without losing its status as a cut set (Rausand (2014)). The **Cut Set Level** is the cardinality of the smallest minimal cut set.

***Definition** 4.* In the fault Analysis domain, the **system loss of function probability** is the occurrence probability of the TOP event defined in the boundaries of interest phase.

The following definition determines how we associate a hypergraph to a system.

***Definition** 5.* Let $\mathcal{S}$ be a system with m independent components $v_1, \ldots v_m$ and n independent failure chains $e_1, \ldots, e_n$ where $e_i = \{v_{i_1}, \ldots, v_{i_s}\}$ is a failure chain. Then the associated hypergraph to the system $\mathcal{S}$ is the hypergraph $\mathcal{H}_{\mathcal{S}} = (V, E)$, where $V = \{v_1, \ldots, v_m\}$ and $E = \{e_1, \ldots, e_n\}$.

As a consequence of the definitions above, we make the following observation how the cut set level and the loss of function probability of a system is connected to its associated hypergraph.

***Observation** 1.* Let $\mathcal{S}$ be a system with m independent components $v_1, \ldots v_m$, n independent failure chains $e_1, \ldots, e_n$, and $p(v_i)$ the loss of function probability of component v_i. Further, let $\mathcal{H}_{\mathcal{S}} = (V, E)$ be the hypergraph associated to the system $\mathcal{S}$. Then the cut set level and system loss of function probability of $\mathcal{S}$ are determined by

$$(i)\ \text{cut\_set\_level}(\mathcal{S}) = \min_{e \in E} |e|,$$
$$(ii)\ \Pr_{loss}(\mathcal{S}) = \sum_{j=1}^{n} \prod_{v \in e_j} p(v).$$

5. EXAMPLE OF HAM SYSTEM ANALYSIS

This section shows the application of the HAM to a system $\mathcal{S}$: a dual motor with a triple-redundancy control system. The system function under analysis is the correct motion. The triple-redundancy consists of three couples of compute & monitor processors. The triple redundancy subsystem output feeds a voter, which means that at least two of the three couples of the redundancy subsystem must be equal to let the system work properly. The voter output drives two motors in parallel, which means that the motors should be in a failure condition concurrently to cause a loss of function. Eventually, a single power-supply powers the triple redundant computer system, but not the backup system or the motors. Fig. 2 shows the dependency diagram for the system. It is worth emphasizing how the system

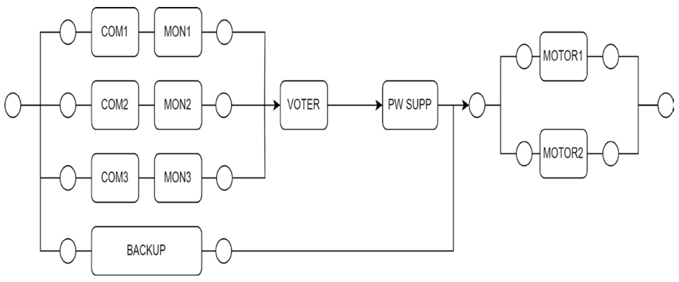


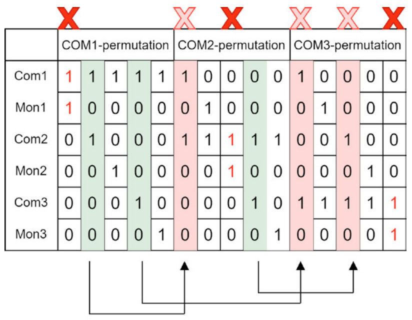
Fig. 2. Triple-redundancy control, dual motors system

under analysis has value only to show the application of the HAM, without being accurate or *realistic*.

The triple-redundancy subsystem is a failure path, and the application of the HAM to such subsystem is an excellent example to show how the HAM helps avoid the duplication of e_i. All the possible failure chains for the redundancy subsystem are the deployments of a pair, one for each redundant branch. The permutation of all components, while one remains selected, is the more straightforward way to build the incidence matrix of the redundancy subsystem hypergraph.

Fig. 3 represents part of the incidence matrix (considering one COM_i fixed while permuting the other objects a time), showing how a bare inspection of the matrix can immediately detect edge duplication and remove them for the incidence matrix before the evaluation phase. Note how edges showing both C_x and M_x set, must be removed as well. After the removal of the duplicates, there will be only twelve significant fault chains for the redundancy subsystem. It is possible to build the Incidence Matrix for the under analysis system according to Fig. 1.

The HAM for the system under analysis has the $\mathcal{H}_\mathcal{S} = (V, E)$ as the hypergraph, where:
$v_1 := COM1$, $v_2 := MON1$, $v_3 := COM2$, $v_4 := MON2$,
$v_5 := COM3$, $v_6 := MON3$, $v_7 := PW.SUPP$, $v_8 := VOTER$,
$v_9 := BACKUP$, $v_{10} := MOTOR1$, $v_{11} := MOTOR2$,
$e_1 = \{COM1, COM2, BACKUP\}$, $e_2 = \{COM1, MON2, BACKUP\}$,
$e_3 = \{COM1, COM3, BACKUP\}$, $e_4 = \{COM1, MON3, BACKUP\}$,
$e_5 = \{MON1, COM2, BACKUP\}$, $e_6 = \{MON1, MON2, BACKUP\}$,
$e_7 = \{MON1, COM3, BACKUP\}$, $e_8 = \{MON1, MON3, BACKUP\}$,
$e_9 = \{COM2, COM3, BACKUP\}$, $e_{10} = \{COM2, MON3, BACKUP\}$,
$e_{11} = \{MON2, COM3, BACKUP\}$, $e_{12} = \{MON2, MON3, BACKUP\}$,
$e_{13} = \{PW.SUPP, BACKUP\}$, $e_{14} = \{VOTER, BACKUP\}$,
$e_{15} = \{MOTOR1, MOTOR2\}$,
$V = \{v_1, \ldots, v_{11}\}$ and $E = \{e_1, \ldots, e_{15}\}$.



	COM1-permutation					COM2-permutation					COM3-permutation				
Com1	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0
Mon1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0
Com2	0	1	0	0	0	1	1	1	1	1	0	0	1	0	0
Mon2	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0
Com3	0	0	0	1	0	0	0	0	1	0	1	1	1	1	1
Mon3	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1

Fig. 3. How to reduce the incidence matrix for the redundancy subsystem

Then Table 2 is the incidence matrix of $\mathcal{H}_\mathcal{S}$ of the system $\mathcal{S}$ in Fig. 2.

Observation 1(i) provides the cut set level calculation of the system,

$$\begin{aligned} \text{cut\_set\_level}(\mathcal{S}) &= \min_{e \in E} |e| = |e_{13}| \\ &= |e_{14}| = |e_{15}| = 2. \end{aligned}$$

Observation 1(ii) provides the loss of function probability for the system,

$$\begin{aligned} &\Pr_{loss}(\mathcal{S}) = \sum_{j=1}^{n} \prod_{v \in e_j} p(v) = \\ &p(v_1)p(v_3)p(v_9) + p(v_1)p(v_4)p(v_9) + \\ &p(v_1)p(v_5)p(v_9) + p(v_1)p(v_6)p(v_9) + \\ &p(v_2)p(v_3)p(v_9) + p(v_2)p(v_4)p(v_9) + \\ &p(v_2)p(v_5)p(v_9) + p(v_2)p(v_6)p(v_9) + \\ &p(v_3)p(v_5)p(v_9) + p(v_3)p(v_6)p(v_9) + \\ &p(v_4)p(v_5)p(v_9) + p(v_4)p(v_6)p(v_9) + \\ &p(v_7)p(v_9) + p(v_8)p(v_9) + p(v_{10})p(v_{11}) \end{aligned} \tag{1}$$

The colors in Table 2 helps to visualize the subsystem contribution to the system and to create a practical example for the hierarchical approach while building the incidence matrix for a complex system, as described in Fig. 1. Without loss of generality, computing and monitoring processors of the redundancy subsystem can have the same failure rate, $p_c = p(v_1) = \ldots = p(v_6)$. Even the two motors can have the same failure, rate $p_{Mt} = p(v_{10}) = p(v_{11})$. With the above assumptions, for a simplified calculation of the total loss of function probability, Equation 1 gives that

$$\Pr_{loss}(\mathcal{S}) = (12p_c^2 + p_P + p_V)p_B + p_{Mt}^2, \tag{2}$$

where $p_P = p(v_7)$, $p_V = p(v_8)$, and $p_B = p(v_9)$.

Note how the loss of function probability calculation doesn't request logic port and logic mathematics skill. After the deductive building of incidence matrix of $\mathcal{H}_\mathcal{S}$,

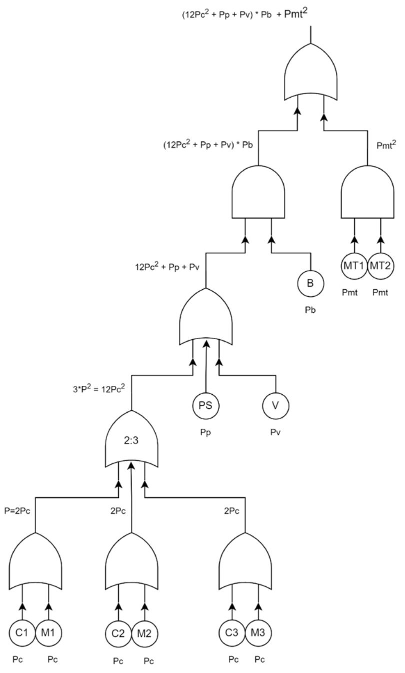
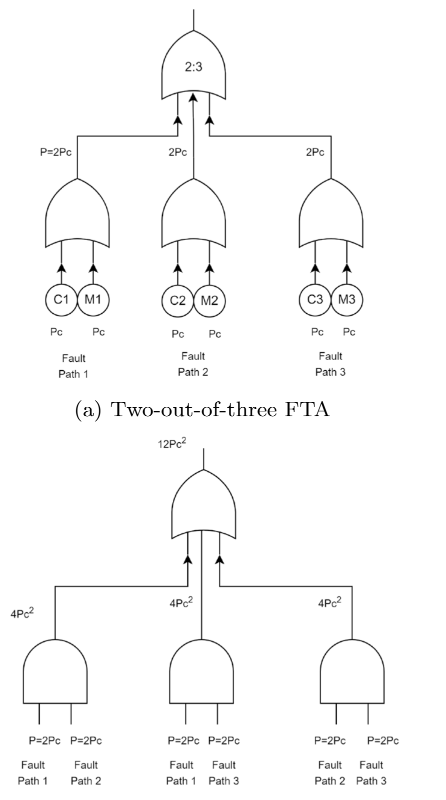


Fig. 4. The FTA method applied to the system in Fig. 2

$V \backslash E$	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{13}	e_{14}	e_{15}
v_1 : COM1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
v_2 : MON1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
v_3 : COM2	1	0	0	0	1	0	0	0	1	1	0	0	0	0	0
v_4 : MON2	0	1	0	0	0	1	0	0	0	0	1	1	0	0	0
v_5 : COM3	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0
v_6 : MON3	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0
v_7 : PW.SUPP	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
v_8 : VOTER	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
v_9 : BACKUP	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
v_{10} : MOTOR1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
v_{11} : MOTOR2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 2. Hypergraph Incidence Matrix for under analysis System (see Fig. 2)

the probability calculation is the simple sum of products per edge e_i. It is possible to validate the HAM procedure by using the FTA method. Fig. 4 shows the result based on the logic gate rules (Ericson (1999)). Shortly, a logic gate OR requests the sum of the probability of the input events, while a logic AND requests the product. 2-out-of-3 logic port is a slightly more complex logic gate. It is the Triple Modular Redundancy (TMR) (Johnson (1989)) with a well known reliability (Dubrova (2013)). The loss of function calculation for a k-out-of-n logic port requests (n k) permutation calculation. A graphical representation for the two-out-of-three port for the system under analysis is possible considering any of couple C-M as a fault path. 2-out-of-3 logic port requires that (at least) two paths are in a fault condition at the same time. Fig. 5 shows the FTA of the two-out-of-three logical working mode. In Fig. 4, per any node, the loss of function probability as calculated applying the logic gate roles is available. Then, the top-level node is the system loss of function probability. It equals Equation 2, validating the HAM procedure.



(a) Two-out-of-three FTA

(b) Two-out-of-three equivalent logic gate roles

Fig. 5. TMR FTA

6. CONCLUSIONS AND FUTURE WORKS

This paper introduces a new deductive method based on the use of hypergraphs to calculate the reliability of a system with bi-stable components. The proposed method removes the need of detailed knowledge of the logic gates and their calculation methodology, and reduces the risk of the duplication of a fault event. The HAM process also applies hierarchically to complex systems. For large systems, the construction of the incidence matrix starts from those of the subsystems with which the system has been partitioned. The HAM process simplifies the calculation of loss-of-function probability

Hypergraphs can be associated with polymatroids via a class of polymatroids called Boolean polymatroids. The concept of Boolean Polymatroids was introduced under the name covering hypermatroids in the 70s (Helgason (1974)). This connection allows us to use polymatroid theory to investigate hypergraphs. As a future work, we want to analyze and mathematically formalize how the characteristics and properties of the polymatroids can automatically reduce even more the risk of duplication of error path during the loss of function calculation. Through the characteristics of polymatroids, the HAM method could provide the means for fault analysis in complex safety systems based on HW and SW components, which is the challenge of traditional fault analysis methods.

REFERENCES

Arnold, F., Belinfante, A., Berg, F.V.D., Guck, D., and Stoelinga, M. (2013). Dftcalc: A tool for efficient fault tree analysis. volume 8153 LNCS. doi:10.1007/978-3-642-40793-2_27.

Aslansefat, K., Kabir, S., Gheraibia, Y., and Papadopoulos, Y. (2020). Dynamic fault tree analysis : State-of-the-art in modeling, analysis, and tools. *Reliability Management and Engineering*.

Berge, C. (1984). *Hypergraphs: Combinatorics of Finite Sets*. North-Holland Mathematical Library. Elsevier Science.

Bhattacharyya, S. and Cheliyan, A. (2019). Optimization of a subsea production system for cost and reliability using its fault tree model. *Reliability Engineering & System Safety*, 185, 213–219.

Biggs, N.L., Lloyd, E.K., and Wilson, R.J. (1998). *Graph Theory 1736-1936*. Clarendon Press, Oxford, England.

Boudali, H., Crouzen, P., and Stoelinga, M. (2010). A rigorous, compositional, and extensible framework for dynamic fault tree analysis. *IEEE Transactions on Dependable and Secure Computing*, 7. doi:10.1109/TDSC.2009.45.

Bouissou, M. and Bon, J.L. (2003). A new formalism that combines advantages of fault-trees and markov models: Boolean logic driven markov processes. *Reliability Engineering and System Safety*, 82. doi:10.1016/S0951-8320(03)00143-1.

Chen, D.J., Mahmud, N., Walker, M., Feng, L., Lönn, H., and Papadopoulos, Y. (2013). Systems modeling with east-adl for fault tree analysis through hip-hops ? volume 4. doi:10.3182/20130904-3-UK-4041.00043.

Dai, Z., Wang, Z., and Jiao, Y. (2011). Dynamic reliability assessment of protection system based on dynamic fault tree and monte carlo simulation. *Zhongguo Dianji Gongcheng Xuebao/Proceedings of the Chinese Society of Electrical Engineering*, 31.

Dubrova, E. (2013). *Fault-Tolerant Design*. Springer New York.

Dugan, J.B., Bavuso, S.J., and Boyd, M.A. (1992). Dynamic fault-tree models for fault-tolerant computer systems. *IEEE Transactions on Reliability*, 41. doi:10.1109/24.159800.

Ericson, C.A. (1999). Fault tree analysis – a history. *The 17th International System Safety Conference*.

Ericson, C.A. (2005). *Hazard Analysis Techniques for System Safety*.

EuroControl (2001). Fault tree analysis (fta) guidance material, edition number 1.0. *European Organisation for the Safety of Air Navigation*, SAF.ET1.ST03.1000-FTA-01-00.

Ferdous, R., Khan, F., Veitch, B., and Amyotte, P. (2007). Methodology for computer-aided fault tree analysis. *Process Safety and Environmental Protection*, 85(1), 70–80.

Guck, D., Spel, J., and Stoelinga, M. (2015). *DFTCalc: Reliability centered maintenance via fault tree analysis (tool paper)*, volume 9407. doi:10.1007/978-3-319-25423-4_19.

Helgason, T. (1974). Aspects of the theory of hypermatroids. In *Hypergraph Seminar*, 191–213. Springer Berlin Heidelberg.

Huang, C.Y. and Chang, Y.R. (2007). An improved decomposition scheme for assessing the reliability of embedded systems by using dynamic fault trees. *Reliability Engineering and System Safety*, 92. doi:10.1016/j.ress.2006.09.008.

Huang, J., You, J.X., Liu, H.C., and Song, M.S. (2020). Failure mode and effect analysis improvement: A systematic literature review and future research agenda. *Reliability Engineering & System Safety*, 199, 106885.

Johnson, B. (1989). *Design and Analysis of Fault-tolerant Digital Systems*. Addison-Wesley series in electrical and computer engineering. Addison-Wesley Publishing Company.

Kabir, S. (2017). An overview of fault tree analysis and its application in model based dependability analysis.

Kabir, S., Walker, M., and Papadopoulos, Y. (2014). Reliability analysis of dynamic systems by translating temporal fault trees into bayesian networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8822. doi:10.1007/978-3-319-12214-4_8.

Khakzad, N., Khan, F., and Amyotte, P. (2013). Dynamic safety analysis of process systems by mapping bow-tie into bayesian network. *Process Safety and Environmental Protection*, 91.

Kritzinger, D. (2017). chapter 4 - fault tree analysis. In D. Kritzinger (ed.), *Aircraft System Safety: Military and Civil Aeronautical Applications*, 59–99. Woodhead Publishing.

Lakehal, A., Nahal, M., and Harouz, R. (2019). Development and application of a decision making tool for fault diagnosis of turbocompressor based on bayesian network and fault tree. *Management and Production Engineering Review*, 10.

Nguyen, T.P., Beugin, J., and Marais, J. (2015). Method for evaluating an extended fault tree to analyse the dependability of complex systems: Application to a satellite-based railway system. *Reliability Engineering and System Safety*, 133. doi:10.1016/j.ress.2014.09.019.

Rao, K.D., Gopika, V., Rao, V.V.S., Kushwaha, H.S., Verma, A.K., and Srividya, A. (2009). Dynamic fault tree analysis using monte carlo simulation in probabilistic safety assessment. *Reliability Engineering and System Safety*, 94. doi:10.1016/j.ress.2008.09.007.

Raptodimos, Y. and Lazakis, I. (2017). Fault tree analysis and artificial neural network modelling for establishing a predictive ship machinery maintenance methodology.

Rausand, M. (2014). *Reliability of Safety-Critical Systems: Theory and Applications*, volume 9781118112724.

Roth, M., Wolf, M., and Lindemann, U. (2015). Integrated matrix-based fault tree generation and evaluation. volume 44.

Ruijters, E. and Stoelinga, M. (2015). Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools. *Computer Science Review*, 15.

Sinnamon, R.M. and Andrews, J.D. (1997). Improved efficiency in qualitative fault tree analysis. *Quality and Reliability Engineering International*, 13.

Vesely, W., Stamatelatos, M., Dugan, J., Fragola, J., III, J.M., and Railsback, J. (2002). Fault tree handbook with aerospace applications version 1.1.

Wang, G., Su, X., and Pan, X. (2012). Computer aided visualized fault tree analysis. In *2012 Fifth International Symposium on Computational Intelligence and Design*, volume 1, 265–268.

Wu, T., Tu, G., Bo, Z.Q., and Klimek, A. (2007). Fuzzy set theory and fault tree analysis based method suitable for fault diagnosis of power transformer. In *2007 International Conference on Intelligent Systems Applications to Power Systems*, 1–5.

Yevkin, O. (2010). An improved monte carlo method in fault tree analysis. doi:10.1109/RAMS.2010.5447989.

Zhu, C. and Zhang, T. (2022). A review on the realization methods of dynamic fault tree. doi:10.1002/qre.3139.

Zhu, P., Han, J., Liu, L., and Zuo, M.J. (2014). A stochastic approach for the analysis of fault trees with priority and gates. *IEEE Transactions on Reliability*, 63. doi:10.1109/TR.2014.2313796.