# MOCUS Algorithm For Obtaining Minimal Cut Sets (MCS) of a Random Computer Network System

Zahir Abdul Haddi Hassan
Manar Mackie Shalaan
Department of Mathematics, College of Education for Pure Sciences University of Babylon
mathzahir@gmail.com

## Abstract

In this paper, the Fault Tree Analysis technique is used in processing a computer network system to: 1) identify basic events or causes of an event: 2) identify common-cause (Minimal cut sets) failures; 3) display causes and consequence of undesired event; 4) evaluate design; and 5) investigate process incidents. The main purpose of fault-tree algorithm is to compute the minimal cut sets as quickly as possible to the calculation of network reliability. A cut set can be defined as a collection of component failure modes that may lead to a system failure. For critical systems, Cut Set Analysis (CSA) is applied to identify and rank system vulnerabilities at design time.

**Keywords:** Reliability Network, Fault Tree Analysis, MOCUS Algorithm for obtaining minimal cut sets , Minimal Cut set.

## I .Introduction

Network reliability has long been a practical problem, and will remain so for years because networks have entered an era of Quality of Service (QoS). IP networks, transportation networks, computer network, mobile phone networks, Nuclear power , electrical power networks, etc., have become "commodities." The goal of telecommunication network operators is to secure Connection availability rates of 99.999%, and premium services may be deployed if the connection reliability is close enough to one. Thus, Reliability is a crucial parameter in the design and analysis of networks[1]. In this paper, we examine a random computer network system of the type two- terminal, the source vertex and the sink vertex .The system can be represented by a probabilistic graph. Consider a complex system in Fig. (1) as a graph $G=(V,E)$, where $V =$ a,b,… ,h . Wherever $a$ and $h$ are the two-terminal of the graph $G$, $E = x_1,x_2,\dots,x_{12}$ [2,3].

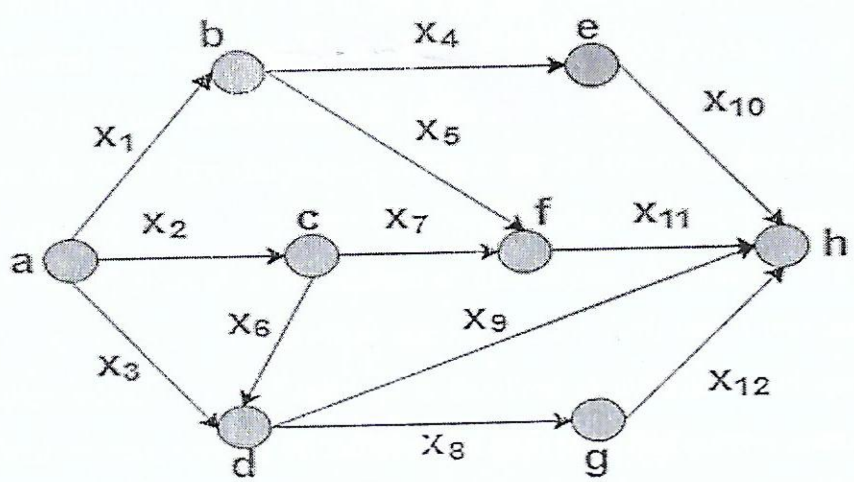



Figure 1: a random of computer network system

The complexity in the design and variation in operating conditions in system require the uncertainty and randomness of fault connected to them [4]. [5] reiterates that the failure of a Computer Network system can lead to environmental detriment and fatal injury to people and economical loss. Analysis of safety computer network system boasts of several techniques which include Fault Tree Analysis (FTA), Failure Mode and Effects (FMEA), Hazard and Operability Analysis (HAZOP) and State Machine Hazard Analysis (SMHA) [6]. Fault Tree Analysis (FTA) has been widely used to determine the reliabilities of complex systems [7]. It has a reputation of being an effective technique to predict probability of hazards caused by a sequence and integration of faults and failure events. It is the most important process for system reliability assessment [8]. It has become a benchmark for safety and reliability of systems. Fault Tree Analysis (FTA) is an important method for the evaluation of complex system's safety and reliability. FTA describes a top-down approach to failure analysis. Basically, being an analytical method, it is used for identifying and classifying hazards and computing reliability of systems [9].The reliability engineer indicates a top event (failure or accident) and then builds the series of faults leading to the top event. It is mostly a logical process for getting combinations of component failures and human errors that could lead to a specified undesired system failure mode [8]. The process starts with an undesirable event known as top-level event and it determines the scenarios which the event may take place in the system. The basic concept in fault tree analysis is the conversion of a physical system into a structured logic diagram (fault tree) in which certain causes lead to one specified TOP event of interest [10]. A Fault Tree depicts a logic diagram that represents certain events that must occur in order for other events to occur. A central focus of fault tree analysis is to find Minimal Cut Sets (MCSs) [11,12]. However, an essential question in fault tree analysis is how many levels of events should be considered. [13] emphasized that obtaining the Minimal cut sets is a necessary step for analyzing a fault tree. The main problem in fault tree analysis is the computation of the minimal cut sets of the fault tree [14] and FTA [15]. [15] argued that the computation of Minimal cut set is NP-hard. [11] described the Minimal Cut Set (MCS) as a substantial concept in fault tree assessment and proposed that they represent a minimal scenario of failure. According to the Fault Tree Handbook, Minimal cut set described the smallest combination of basic events that cause a top event. Likewise, Fault Tree Handbook stated that successful computation of Minimal Cut Set (MCS) simplified the process of quantifying the process of Fault trees[16]. The process involved in computing the Minimal cut sets for smaller Fault tree is less boring. However, the task becomes complex when a larger Fault tree is involved as algorithm and codes are required to determine the minimal cut sets, making effective use of fault trees that resolve causes of system failure relies on the fault tree algorithm that provides optimal cut sets. This paper provides understanding on how of the fault tree algorithm carries out the computation of minimal cut sets of complex fault trees.

## II. Basic Concepts

**1.Definition**. A graph $G = (V, E)$ consists of set $V$ of vertices (nodes) and set E of edges (links).

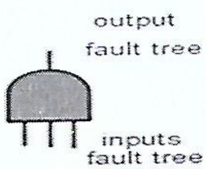
**2.Definition.** Terminal Reliability The two terminal reliability is the probability that a communication path must exist between a specific pair of nodes of a network under a predefined working environment.

**3. Definition. A** cut set in a fault tree is a set of basic events whose occurrence (at the same time) ensures that the TOP event occurs.

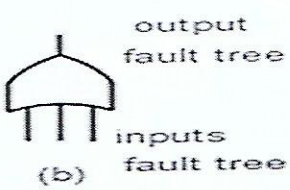
**4. Definition.** A cut set is said to be minimal if the set cannot be reduced without loosing its status **as** a cut set.

**5. Definition. Fault Tree Analysis (FTA)**

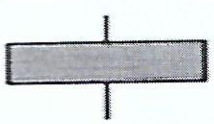
The fault tree is a logic diagram based on the principle of multi-causality, which traces all branches of events which could contribute to an accident or failure. It uses sets of symbols, labels and identifiers. But for our purposes, we only use a handful of these, shown below:



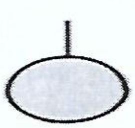

(a) the AND gate denotes that an output fault event occurs only if all the input fault events occur.




(b) The OR gate denotes that an output fault occurs if one or more of the input fault events occur.



(c) The rectangle denotes a fault event that results from the combination of fault events through the input of a logic gate.



(d) The circle denotes a basic fault event or the failure of a basic component. The event's occurrence probability and failure and repair rates are normally obtained from empirical data.

## III.MOCUS Algorithm For Obtaining Minimal Cut Sets (MCS)

One of the challenging problems confronting the fault tree method is obtaining minimal cut sets or eliminating repeated events of a fault tree. This section presents one algorithm to obtain minimal cut sets of a fault tree [12,17-20].A *cut set* may be described as a collection of basic events that will cause the top fault tree event to occur. Additionally, a cut set is said to be minimal if it cannot be further minimized or reduced but it can still ensure the occurrence of the top fault tree event. The algorithm in question can either be used manually for simple fault trees or computerized for complex fault trees with hundreds of gates and basic fault events. In this algorithm, the AND gate increases the size of a cut set and the OR gate increases the number of cut sets. The algorithm is demonstrated by solving the following example

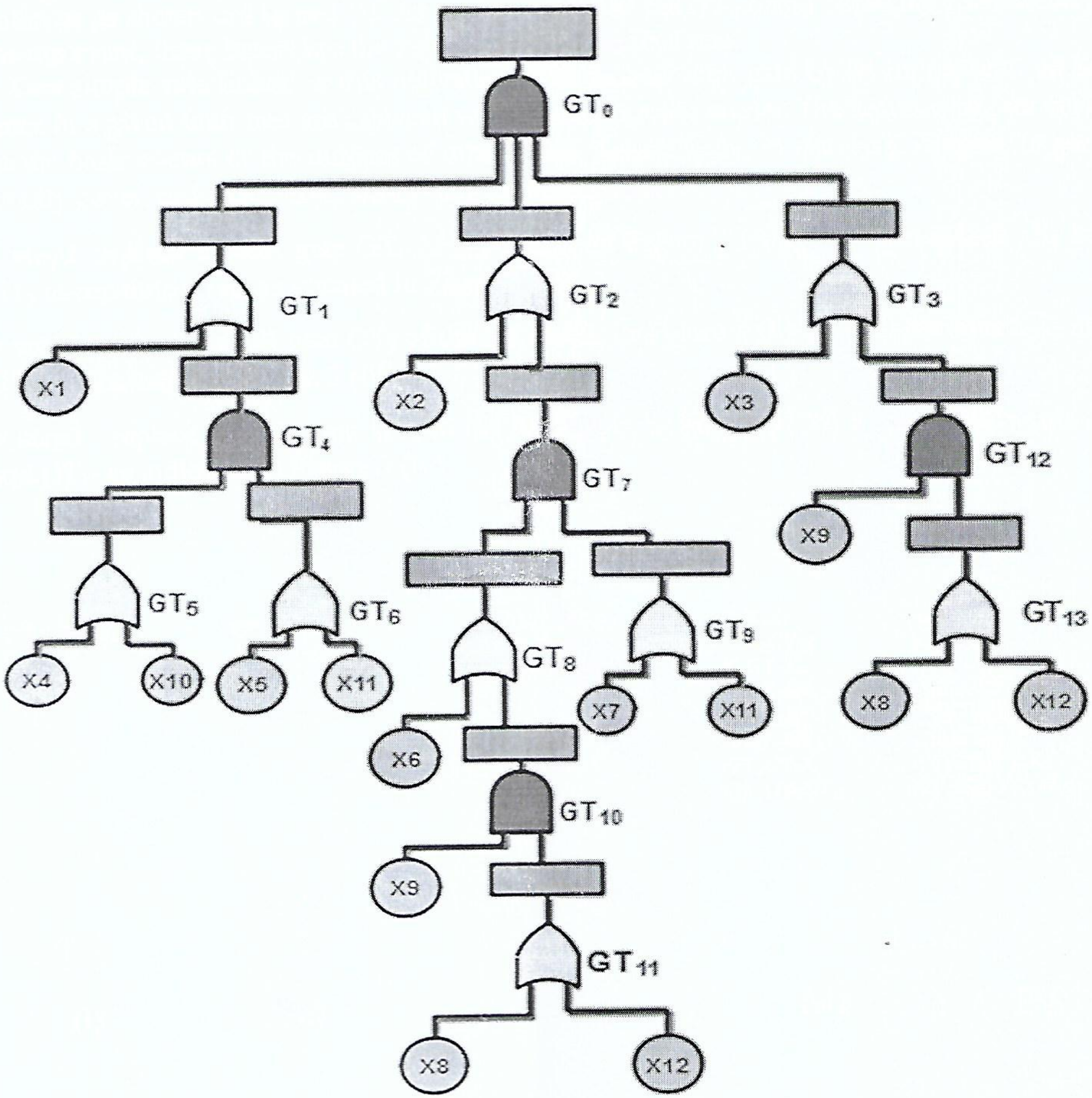



**Figure2:**An event tree with a repeated event.

# Example

Obtain a repeated event free fault tree of the fault tree shown in Figure2. In this figure, the basic fault events are identified by the numerals and the logic gates are labeled as $GT_0$, $GT_1$, $GT_2$, …, $GT_{13}$.

The algorithm begins from the gate, $GT_0$, just below the top event of the fault tree shown in Figure 2. Normally, in a fault tree this gate is either OR or AND. If this gate is an AND gate, then each of its inputs dentes an entry for each column of the list matrix. On the other hand, if this gate is an OR, then each of its inputs represents an entry for each row of the list matrix.

In our case, $GT_0$ is an AND gate; therfore, we start the formulation of the list matrix by listing the gate inputs: $GT_1$, $GT_2$, and $GT_3$ (output events) in a single row but in separate columns as shown in Figure3 (i). As any one input of the $GT_0$ could cause the occurrence of the top event, these inputs are the members of distinct cut sets.

One simple rule linked to this algorithm is to replace each gate by its inputs until all the gates in a given fault tree are replaced with the basic event entries.The inputs of a gate could be the basic events or the outputs of other gates. Accordingly, in our case, in order to get a wholly developed list matrix, we use the following steps:

◆ **step1** replace the OR gate $GT_1$ of list matrix of Figure 3 (i) by its input events $X_1$ and $GT_4$ as separate rows, as idicated in Figure 3 (ii).

◆ **step2** replace the OR gate $GT_2$ of the list matrix in Figure 3 (ii) by its input events $X_2$ and $GT_7$, as indicated in Figure 3 (iii).

◆ **step3** replace the OR gate $GT_3$ of the list matrix in Figure 3 (iii) by its input events $X_3$ and $GT_{12}$ as indicated in Figure 3 (iv).

| | | | | |
|---|---|---|---|---|
| $GT_1GT_2GT_3$ | $GT_1GT_2GT_3$ | $X_1GT_7GT_3$ | $X_1X_2X_3$ | $X_1X_2X_3$ |
| | $GT_4GT_2GT_3$ | $GT_4X_2GT_3$ | $X_1X_2GT_{12}$ | $X_1X_2GT_{12}$ |
| | | $GT_4GT_7GT_3$ | $X_1GT_7X_3$ | $X_1GT_7X_3$ |
| | | | $X_1GT_7GT_{12}$ | $X_1GT_7GT_{12}$ |
| | | | $GT_4X_2X_3$ | $GT_5GT_6X_2X_3$ |
| | | | $GT_4X_2GT_{12}$ | $GT_5GT_6X_2GT_{12}$ |
| | | | $GT_4GT_7X_3$ | $GT_5GT_6GT_7X_3$ |
| | | | $GT_4GT_7GT_{12}$ | $GT_5GT_6GT_7GT_{12}$ |
| **(i)** | **(ii)** | **(iii)** | **(iv)** | **(v)** |

$X_1X_2X_3$
$X_1X_2GT_{12}$
$X_1GT_7X_3$
$X_1GT_7GT_{12}$
$X_4GT_6X_2X_3$
$X_{10}GT_6X_2X_3$
$X_4GT_6X_2GT_{12}$
$X_{10}GT_6X_2GT_{12}$
$X_4GT_6GT_7X_3$
$X_{10}GT_6GT_7X_3$
$X_4GT_6GT_7GT_{12}$
$X_{10}GT_6GT_7GT_{12}$

(vi)

$X_1X_2X_3$
$X_1X_2GT_{12}$
$X_1GT_7X_3$
$X_1GT_7GT_{12}$
$X_4X_5X_2X_3$
$X_4X_{11}X_2X_3$
$X_{10}X_5X_2X_3$
$X_{10}X_{11}X_2X_3$
$X_4X_5X_2GT_{12}$
$X_4X_{11}X_2GT_{12}$
$X_{10}X_5X_2GT_{12}$
$X_{10}X_{11}X_2GT_{12}$
$X_4X_5GT_7X_3$
$X_4X_{11}GT_7X_3$
$X_{10}X_5GT_7X_3$
$X_{10}X_{11}GT_7X_3$
$X_4X_5GT_7GT_{12}$
$X_4X_{11}GT_7GT_{12}$
$X_{10}X_5GT_7GT_{12}$
$X_{10}X_{11}GT_7GT_{12}$

(vii)

$X_1X_2X_3$
$X_1X_2GT_{12}$
$X_1GT_8GT_9X_3$
$X_1GT_8GT_9GT_{12}$
$X_4X_5X_2X_3$
$X_4X_{11}X_2X_3$
$X_{10}X_5X_2X_3$
$X_{10}X_{11}X_2X_3$
$X_4X_5X_2GT_{12}$
$X_4X_{11}X_2GT_{12}$
$X_{10}X_5X_2GT_{12}$
$X_{10}X_{11}X_2GT_{12}$
$X_4X_5GT_8GT_9X_3$
$X_4X_{11}GT_8GT_9X_3$
$X_{10}X_5GT_8GT_9X_3$
$X_{10}X_{11}GT_8GT_9X_3$
$X_4X_5GT_8GT_9GT_{12}$
$X_4X_{11}GT_8GT_9GT_{12}$
$X_{10}X_5GT_8GT_9GT_{12}$
$X_{10}X_{11}GT_8GT_9GT_{12}$

(viii)

$X_1X_2X_3$
$X_1X_2GT_{12}$
$X_1X_6GT_9X_3$
$X_1GT_{10}GT_9X_3$
$X_1X_6GT_9GT_{12}$
$X_1GT_{10}GT_9GT_{12}$
$X_4X_5X_2X_3$
$X_4X_{11}X_2X_3$
$X_{10}X_5X_2X_3$
$X_{10}X_{11}X_2X_3$
$X_4X_5X_2GT_{12}$
$X_4X_{11}X_2GT_{12}$
$X_{10}X_5X_2GT_{12}$
$X_{10}X_{11}X_2GT_{12}$
$X_4X_5X_6GT_9X_3$
$X_4X_5GT_{10}GT_9X_3$
$X_4X_{11}X_6GT_9X_3$
$X_4X_{11}GT_{10}GT_9X_3$
$X_{10}X_5X_6GT_9X_3$
$X_{10}X_5GT_{10}GT_9X_3$
$X_{10}X_{11}X_6GT_9X_3$
$X_{10}X_{11}GT_{10}GT_9X_3$
$X_4X_5X_6GT_9GT_{12}$
$X_4X_5GT_{10}GT_9GT_{12}$
$X_4X_{11}X_6GT_9GT_{12}$
$X_4X_{11}GT_{10}GT_9GT_{12}$
$X_{10}X_5X_6GT_9GT_{12}$
$X_{10}X_5GT_{10}GT_9GT_{12}$
$X_{10}X_{11}X_6GT_9GT_{12}$
$X_{10}X_{11}GT_{10}GT_9GT_{12}$

(ix)

$X_1X_2X_3$
$X_1X_2GT_{12}$
$X_1X_6X_7X_3$
$X_1X_6X_{11}X_3$
$X_1GT_{10}X_7X_3$
$X_1GT_{10}X_{11}X_3$
$X_1X_6X_7GT_{12}$
$X_1X_6X_{11}GT_{12}$
$X_1G_{10}X_7GT_{12}$
$X_1GT_{10}X_{11}GT_{12}$
$X_4X_5X_2X_3$
$X_4X_{11}X_2X_3$
$X_{10}X_5X_2X_3$
$X_{10}X_{11}X_2X_3$
$X_4X_5X_2GT_{12}$
$X_4X_{11}X_2GT_{12}$
$X_{10}X_5X_2GT_{12}$
$X_{10}X_{11}X_2GT_{12}$
$X_4X_5X_6X_7X_3$
$X_4X_5X_6X_{11}X_3$
$X_4X_5GT_{10}X_7X_3$
$X_4X_5GT_{10}X_{11}X_3$
$X_4X_{11}X_6X_7X_3$
$X_4X_{11}X_6X_{11}X_3$
$X_4X_{11}GT_{10}X_7X_3$
$X_4X_{11}GT_{10}X_3$
$X_{10}X_5X_6X_7X_3$
$X_{10}X_5X_6X_{11}X_3$
$X_{10}X_5GT_{10}X_7X_3$
$X_{10}X_5GT_{10}X_{11}X_3$
$X_{10}X_{11}X_6\ X_7X_3$
$X_{10}X_{11}X_6X_3$
$X_{10}X_{11}GT_{10}X_7X_3$
$X_{10}X_{11}GT_{10}X_3$
$X_4X_5X_6X_7GT_{12}$
$X_4X_5X_6X_{11}GT_{12}$
$X_4X_5GT_{10}X_7GT_{12}$
$X_4X_5GT_{10}X_{11}GT_{12}$
$X_4X_{11}X_6X_7GT_{12}$
$X_4X_{11}X_6GT_{12}$
$X_4X_{11}GT_{10}X_7GT_{12}$
$X_4X_{11}GT_{10}GT_{12}$
$X_{10}X_5X_6X_7GT_{12}$
$X_{10}X_5X_6X_{11}GT_{12}$
$X_{10}X_5GT_{10}X_7GT_{12}$
$X_{10}X_5GT_{10}X_{11}GT_{12}$
$X_{10}X_{11}X_6X_7GT_{12}$
$X_{10}X_{11}X_6GT_{12}$
$X_{10}X_{11}GT_{10}X_7GT_{12}$
$X_{10}X_{11}GT_{10}GT_{12}$

(x)

$X_1X_2X_3$
$X_1X_2GT_{12}$
$X_1X_6X_7X_3$
$X_1X_6X_{11}X_3$
$X_1GT_{11}X_7X_3X_9$
$X_1GT_{11}X_{11}X_3X_9$
$X_1X_6X_7GT_{12}$
$X_1X_6X_{11}GT_{12}$
$X_1GT_{11}X_7GT_{12}X_9$
$X_1GT_{11}X_{11}GT_{12}X_9$
$X_4X_5X_2X_3$
$X_4X_{11}X_2X_3$
$X_{10}X_5X_2X_3$
$X_{10}X_{11}X_2X_3$
$X_4X_5X_2GT_{12}$
$X_4X_{11}X_2GT_{12}$
$X_{10}X_5X_2GT_{12}$
$X_{10}X_{11}X_2GT_{12}$
$X_4X_5X_6X_7X_3$
$X_4X_5X_6X_{11}X_3$
$X_4X_5GT_{11}X_7X_3X_9$
$X_4X_5GT_{11}X_{11}X_3X_9$
$X_4X_{11}X_6X_7X_3$
$X_4X_{11}X_6X_3$
$X_4X_{11}GT_{11}X_7X_3X_9$
$X_4X_{11}GT_{11}X_3X_9$
$X_{10}X_5X_6X_7X_3$
$X_{10}X_5X_6X_{11}X_3$
$X_{10}X_5GT_{11}X_7X_3X_9$
$X_{10}X_5GT_{11}X_{11}X_3X_9$
$X_{10}X_{11}X_6X_7X_3$
$X_{10}X_{11}X_6X_3$
$X_{10}X_{11}GT_{11}X_7X_3X_9$
$X_{10}X_{11}GT_{11}X_3X_9$
$X_4X_5X_6X_7GT_{12}$
$X_4X_5X_6X_{11}GT_{12}$
$X_4X_5GT_{11}X_7GT_{12}X_9$
$X_4X_5GT_{11}X_{11}GT_{12}X_9$
$X_4X_{11}X_6X_7GT_{12}$
$X_4X_{11}X_6GT_{12}$
$X_4X_{11}GT_{11}X_7GT_{12}X_9$
$X_4X_{11}GT_{11}GT_{12}X_9$
$X_{10}X_5X_6X_7GT_{12}$
$X_{10}X_5X_6X_{11}GT_{12}$
$X_{10}X_5GT_{11}X_7GT_{12}X_9$
$X_{10}X_5GT_{11}X_{11}GT_{12}X_9$
$X_{10}X_{11}X_6X_7GT_{12}$
$X_{10}X_{11}X_6GT_{12}$
$X_{10}X_{11}GT_{11}X_7GT_{12}X_9$
$X_{10}X_{11}GT_{11}GT_{12}X_9$

(xi)

$X_1X_2X_3$
$X_1X_2GT_{12}$
$X_1X_6X_7X_3$
$X_1X_6X_{11}X_3$
$X_1X_7X_3X_9X_8$
$X_1X_7X_3X_9X_{12}$
$X_1X_{11}X_3X_9X_8$
$X_1X_{11}X_3X_9X_{12}$
$X_1X_6X_7GT_{12}$
$X_1X_6X_{11}GT_{12}$
$X_1X_7GT_{12}X_9X_8$
$X_1X_7GT_{12}X_9X_{12}$
$X_1X_{11}GT_{12}X_9X_8$
$X_1X_{11}GT_{12}X_9X_{12}$
$X_4X_5X_2X_3$
$X_4X_{11}X_2X_3$
$X_{10}X_5X_2X_3$
$X_{10}X_{11}X_2X_3$
$X_4X_5X_2GT_{12}$
$X_4X_{11}X_2GT_{12}$
$X_{10}X_5X_2GT_{12}$
$X_{10}X_{11}X_2GT_{12}$
$X_4X_5X_6X_7X_3$
$X_4X_5X_6X_{11}X_3$
$X_4X_5X_7X_3X_9X_8$
$X_4X_5X_7X_3X_9X_{12}$
$X_4X_5X_{11}X_3X_9X_8$
$X_4X_5X_{11}X_3X_9X_{12}$
$X_4X_{11}X_6X_7X_3$
$X_4X_{11}X_6X_3$
$X_4X_{11}X_7X_3X_9X_8$
$X_4X_{11}X_7X_3X_9X_{12}$
$X_4X_{11}X_3X_9X_8$
$X_4X_{11}X_3X_9X_{12}$
$X_{10}X_5X_6X_7X_3$
$X_{10}X_5X_6X_{11}X_3$
$X_{10}X_5X_7X_3X_9X_8$
$X_{10}X_5X_7X_3X_9X_{12}$
$X_{10}X_5X_{11}X_3X_9X_8$
$X_{10}X_5X_{11}X_3X_9X_{12}$
$X_{10}X_{11}X_6X_7X_3$
$X_{10}X_{11}X_6X_3$
$X_{10}X_{11}X_8X_7X_3X_9$
$X_{10}X_{11}X_{12}X_7X_3X_9$
$X_{10}X_{11}X_8X_3X_9$
$X_{10}X_{11}X_{12}X_3X_9$
$X_4X_5X_6X_7GT_{12}$
$X_4X_5X_6X_{11}GT_{12}$
$X_4X_5X_8X_7GT_{12}X_9$
$X_4X_5X_{12}X_7GT_{12}X_9$
$X_4X_5X_8X_{11}GT_{12}X_9$
$X_4X_5X_{12}X_{11}GT_{12}X_9$
$X_4X_{11}X_6X_7GT_{12}$
$X_4X_{11}X_6GT_{12}$
$X_4X_{11}X_8X_7GT_{12}X_9$
$X_4X_{11}X_{12}X_7GT_{12}X_9$
$X_4X_{11}X_8GT_{12}X_9$
$X_4X_{11}X_{12}GT_{12}X_9$
$X_{10}X_5X_6X_7GT_{12}$
$X_{10}X_5X_6X_{11}GT_{12}$
$X_{10}X_5X_8X_7GT_{12}X_9$
$X_{10}X_5X_{12}X_7GT_{12}X_9$
$X_{10}X_5X_8X_{11}GT_{12}X_9$
$X_{10}X_5X_{12}X_{11}GT_{12}X_9$
$X_{10}X_{11}X_6X_7GT_{12}$
$X_{10}X_{11}X_6GT_{12}$
$X_{10}X_{11}X_8X_7GT_{12}X_9$
$X_{10}X_{11}X_{12}X_7GT_{12}X_9$
$X_{10}X_{11}X_8GT_{12}X_9$
$X_{10}X_{11}X_{12}GT_{12}X_9$

(xii)

$X_1X_2X_3$
$X_1X_2X_9GT_{13}$
$X_1X_6X_7X_3$
$X_1X_6X_{11}X_3$
$X_1X_7X_3X_9X_8$
$X_1X_7X_3X_9X_{12}$
$X_1X_{11}X_3X_9X_8$
$X_1X_{11}X_3X_9X_{12}$
$X_1X_6X_7X_9GT_{13}$
$X_1X_6X_{11}X_9GT_{13}$
$X_1X_7X_8X_9GT_{13}$
$X_1X_7X_9X_{12}GT_{13}$
$X_1X_{11}X_9GT_{13}X_8$
$X_1X_{11}X_9GT_{13}X_{12}$
$X_4X_5X_2X_3$
$X_4X_{11}X_2X_3$
$X_{10}X_5X_2X_3$
$X_{10}X_{11}X_2X_3$
$X_4X_5X_2X_9GT_{13}$
$X_4X_{11}X_2X_9GT_{13}$
$X_{10}X_5X_2X_9GT_{13}$
$X_{10}X_{11}X_2X_9GT_{13}$
$X_4X_5X_6X_7X_3$
$X_4X_5X_6X_{11}X_3$
$X_4X_5X_7X_3X_9X_8$
$X_4X_5X_7X_3X_9X_{12}$
$X_4X_5X_{11}X_3X_9X_8$
$X_4X_5X_{11}X_3X_9X_{12}$
$X_4X_{11}X_6X_7X_3$
$X_4X_{11}X_6X_3$
$X_4X_{11}X_7X_3X_9X_8$
$X_4X_{11}X_7X_3X_9X_{12}$
$X_4X_{11}X_3X_9X_8$
$X_4X_{11}X_3X_9X_{12}$
$X_{10}X_5X_6X_7X_3$
$X_{10}X_5X_6X_{11}X_3$
$X_{10}X_5X_7X_3X_9X_8$
$X_{10}X_5X_7X_3X_9X_{12}$
$X_{10}X_5X_{11}X_3X_9X_8$
$X_{10}X_5X_{11}X_3X_9X_{12}$
$X_{10}X_{11}X_6X_7X_3$
$X_{10}X_{11}X_6X_3$
$X_{10}X_{11}X_8X_7X_3X_9$
$X_{10}X_{11}X_{12}X_7X_3X_9$
$X_{10}X_{11}X_8X_3X_9$
$X_{10}X_{11}X_{12}X_3X_9$
$X_4X_5X_6X_7X_9GT_{13}$
$X_4X_5X_6X_{11}X_9GT_{13}$
$X_4X_5X_8X_7X_9GT_{13}$
$X_4X_5X_{12}X_7X_9GT_{13}$
$X_4X_5X_8X_{11}X_9GT_{13}$
$X_4X_5X_{12}X_{11}X_9GT_{13}$
$X_4X_{11}X_6X_7X_9GT_{13}$
$X_4X_{11}X_6X_9GT_{13}$
$X_4X_{11}X_8X_7X_9GT_{13}$
$X_4X_{11}X_{12}X_7X_9GT_{13}$
$X_4X_{11}X_8X_9GT_{13}$
$X_4X_{11}X_{12}X_9GT_{13}$
$X_{10}X_5X_6X_7X_9GT_{13}$
$X_{10}X_5X_6X_{11}X_9GT_{13}$
$X_{10}X_5X_8X_7X_9GT_{13}$
$X_{10}X_5X_{12}X_7X_9GT_{13}$
$X_{10}X_5X_8X_{11}X_9GT_{13}$
$X_{10}X_5X_{12}X_{11}X_9GT_{13}$
$X_{10}X_{11}X_6X_7X_9GT_{13}$
$X_{10}X_{11}X_6X_9GT_{13}$
$X_{10}X_{11}X_8X_7X_9GT_{13}$
$X_{10}X_{11}X_{12}X_7X_9GT_{13}$
$X_{10}X_{11}X_8X_9GT_{13}$
$X_{10}X_{11}X_{12}X_9GT_{13}$

(xiii)

$X_1X_2X_3$
$X_1X_2X_8X_9$
$X_1X_2X_9X_{12}$
$X_1X_3X_6X_7$
$X_1X_3X_6X_{11}$
$X_1X_7X_8X_9$
$X_1X_7X_9X_{12}$
$X_1X_8X_9X_{11}$
$X_2X_3X_4X_5$
$X_2X_3X_4X_{11}$
$X_2X_3X_5X_{10}$
$X_2X_3X_{10}X_{11}$
$X_1X_9X_{11}X_{12}$
$X_3X_4X_6X_{11}$
$X_3X_6X_{10}X_{11}$
$X_4X_8X_9X_{11}$
$X_4X_9X_{11}X_{12}$
$X_8X_9X_{10}X_{11}$
$X_9X_{10}X_{11}X_{12}$
$X_1X_3X_7X_8X_9$
$X_1X_3X_7X_9X_{12}$
$X_1X_3X_8X_9X_{11}$
$X_1X_3X_9X_{11}X_{12}$
$X_1X_6X_7X_9X_8$
$X_1X_6X_7X_9X_{12}$
$X_1X_6X_{11}X_9X_8$
$X_1X_6X_{11}X_9X_{12}$
$X_1X_7X_8X_9X_{12}$
$X_1X_7X_8X_9X_{12}$
$X_1X_{11}X_9X_8X_{12}$
$X_1X_{11}X_9X_8X_{12}$
$X_2X_4X_5X_8X_9$
$X_2X_4X_5X_9X_{12}$
$X_2X_4X_8X_9X_{11}$
$X_2X_4X_9X_{11}X_{12}$
$X_2X_5X_8X_9X_{10}$
$X_2X_5X_9X_{10}X_{12}$
$X_2X_8X_9X_{10}X_{11}$
$X_2X_9X_{10}X_{11}X_{12}$
$X_3X_4X_5X_6X_7$
$X_3X_4X_5X_6X_{11}$
$X_3X_4X_5X_7X_8X_9$
$X_3X_4X_5X_7X_9X_{12}$
$X_3X_4X_5X_8X_9X_{11}$
$X_3X_4X_5X_9X_{11}X_{12}$
$X_3X_4X_6X_7X_{11}$
$X_3X_4X_7X_8X_9X_{11}$
$X_3X_4X_7X_9X_{11}X_{12}$
$X_3X_4X_8X_9X_{11}$
$X_3X_4X_9X_{11}X_{12}$
$X_3X_5X_6X_7X_{10}$
$X_3X_5X_6X_{10}X_{11}$
$X_{10}X_5X_7X_3X_9X_8$
$X_{10}X_5X_7X_3X_9X_{12}$
$X_{10}X_5X_{11}X_3X_9X_8$
$X_{10}X_5X_{11}X_3X_9X_{12}$
$X_{10}X_{11}X_6X_7X_3$
$X_{10}X_{11}X_6X_3$
$X_{10}X_{11}X_8X_7X_3X_9$
$X_{10}X_{11}X_{12}X_7X_3X_9$
$X_{10}X_{11}X_8X_3X_9$
$X_{10}X_{11}X_{12}X_3X_9$
$X_4X_5X_6X_7X_9X_8$
$X_4X_5X_6X_7X_9X_{12}$
$X_4X_5X_6X_{11}X_9X_8$
$X_4X_5X_6X_{11}X_9X_{12}$
$X_4X_5X_7X_8X_9$
$X_4X_5X_8X_7X_9X_{12}$
$X_4X_5X_{12}X_7X_9X_8$
$X_4X_5X_7X_9X_{12}$
$X_4X_5X_8X_9X_{11}$
$X_4X_5X_8X_{11}X_9X_{12}$
$X_4X_5X_{12}X_{11}X_9X_8$
$X_4X_5X_9X_{11}X_{12}$
$X_4X_{11}X_6X_7X_9X_8$
$X_4X_{11}X_6X_7X_9X_{12}$
$X_4X_{11}X_6X_9X_8$
$X_4X_{11}X_6X_9X_{12}$
$X_4X_{11}X_8X_7X_9$
$X_4X_{11}X_8X_7X_9X_{12}$
$X_4X_{11}X_{12}X_7X_9X_8$
$X_4X_{11}X_{12}X_7X_9$
$X_4X_{11}X_8X_9X_{12}$
$X_4X_{11}X_{12}X_9X_8$
$X_{10}X_5X_6X_7X_9X_8$
$X_{10}X_5X_6X_7X_9X_{12}$
$X_{10}X_5X_6X_{11}X_9X_8$
$X_{10}X_5X_6X_{11}X_9X_{12}$
$X_5X_7X_8X_9X_{10}$
$X_{10}X_5X_8X_7X_9X_{12}$
$X_{10}X_5X_{12}X_7X_9X_8$
$X_{10}X_5X_{12}X_7X_9$
$X_{10}X_5X_8X_{11}X_9$
$X_{10}X_5X_8X_{11}X_9X_{12}$
$X_{10}X_5X_{12}X_{11}X_9X_8$
$X_{10}X_5X_{12}X_{11}X_9$
$X_{10}X_{11}X_6X_7X_9X_8$
$X_{10}X_{11}X_6X_7X_9X_{12}$
$X_{10}X_{11}X_6X_9X_8$
$X_{10}X_{11}X_6X_9X_{12}$
$X_{10}X_{11}X_8X_7X_9$
$X_{10}X_{11}X_8X_7X_9X_{12}$
$X_{10}X_{11}X_{12}X_7X_9X_8$
$X_{10}X_{11}X_{12}X_7X_9$
$X_{10}X_{11}X_8X_9X_{12}$
$X_{10}X_{11}X_{12}X_9X_8$

(xiv)

$X_1,X_2,X_3$
$X_1,X_2,X_8,X_9$
$X_1,X_2,X_9,X_2$
$X_1,X_3,X_6,X_7$
$X_1,X_3,X_6,X_{11}$
$X_1,X_7,X_8,X_9$
$X_1,X_7,X_9,X_{12}$
$X_1X_8X_9X_{11}$
$X_1,X_9,X_{11},X_{12}$
$X_2,X_3,X_4,X_5$
$X_2,X_3,X_4,X_{11}$
$X_2,X_3,X_5,X_{10}$
$X_2,X_3,X_{10},X_{11}$
$X_3$, X4, X6, $X_{11}$
$X_3,X_6,X_{10},X_{11}$
$X_4,X_8,X_9,X_{11}$
$X_4,X_9,X_{11},X_{12}$
$X_8,X_9,X_{10},X_{11}$
$X_9,X_{10},X_{11},X_{12}$
$X_2X_4X_5X_8X_9$
$X_2,X_4,X_5,X_9,X_{12}$
$X_2,X_5,X_8,X_9,X_{10}$
$X_2,X_5,X_9,X_{10},X_{12}$
$X_3,X_4,X_5,X_6,X_7$
$X_3,X_5,X_6,X_7,X_{10}$
$X_4,X_5,X_7,X_8,X_9$
$X_4,X_5,X_7,X_9,X_{12}$
$X_4,X_5,X_8,X_9,X_{11}$
$X_5,X_7,X_8,X_9,X_{10}$

(xv)

**Figure3:** List matrix under different conditions.

♦ **step4** replace the AND gate $GT_4$ of the list matrix in Figure 3(iv) by its input events $GT_5$ and $GT_6$, as indicated in Figure 3(v).

♦ **step5** replace the OR gate $GT_5$ of the list matrix in Figure 3(v) by its input events $X_4$ and $X_{10}$, as indicated in Figure 3(vi).

♦ **step6** replace the OR gate $GT_6$ of the list matrix in Figure 3(vi) by its input events $X_5$ and $X_{11}$, as indicated in Figure 3(vii).

♦ **step7** replace the AND gate $GT_7$ of the list matrix in Figure 3(vii) by its input events $GT_8$ and $GT_9$, as indicated in Figure 3(viii).

♦ **step8** replace the OR gate $GT_8$ of the list matrix in Figure 3(viii) by its input events $X_6$ and $GT_{10}$, as indicated in Figure 3(ix).

♦ **step9** replace the OR gate $GT_9$ of the list matrix in Figure 3(ix) by its input events $X_7$ and $X_{11}$, as indicated in Figure 3(x).

♦ **step10** replace the AND gate $GT_{10}$ of the list matrix in Figure 3(x) by its input events $X_9$ and $GT_{11}$, as indicated in Figure 3(xi).

**step11** replace the OR gate $GT_{11}$ of the list matrix in Figure 3(xi) by its input events $X_8$ and $X_{12}$, as indicated in Figure 3(xii).

**step12** replace the AND gate $GT_{12}$ of the list matrix in Figure 3(xii) by its input events $X_9$ and $GT_{13}$, as indicated in Figure 3(xiii).

**step13** replace the OR gate $GT_{13}$ of the list matrix in Figure 3(xiii) by its input events $X_8$ and $X_{12}$, as indicated in Figure 3(xiv).

As shown in the list matrix of Figure 3(xiv),the cut sets $\{X_1,X_7,X_8,X_9\},\{X_1,X_7,X_9,X_{12}\},\{X_1X_8X_9X_{11}\},\{X_1,X_9,X_{11},X_{12}\},\ldots$ events cut sets. As only its occurrence will result in the occurrence of the top event, we eliminate cut sets $\{X_1,X_6,X_7,X_8,X_9\}$, $\{X_1,X_7,X_8,X_9,X_{12}\}$,$\{X_1,X_6,X_7,X_9,X_{12}\}$ ,$\{X_1,X_6,X_8,X_9,X_{11}\}$, $\{X_1,X_6,X_9,X_{11},X_{12}\},\ldots$ from the list matrix of Figure 3(xiv), because the occurrence of this cut set requires all events $X_1$, $X_6$, $X_7$,$X_8$ and $X_9$ to occur, Similarly for the list cut sets. Consequently, the list matrix shown in Figure3 (xv) represents the minimal cut sets of the fault tree given in Figure 2.The fault tree of the Figure 3 (xv) list matrix is shown in Figure 4. Now this fault tree can be used to obtain the quantitative measures of the top or undesirable event.

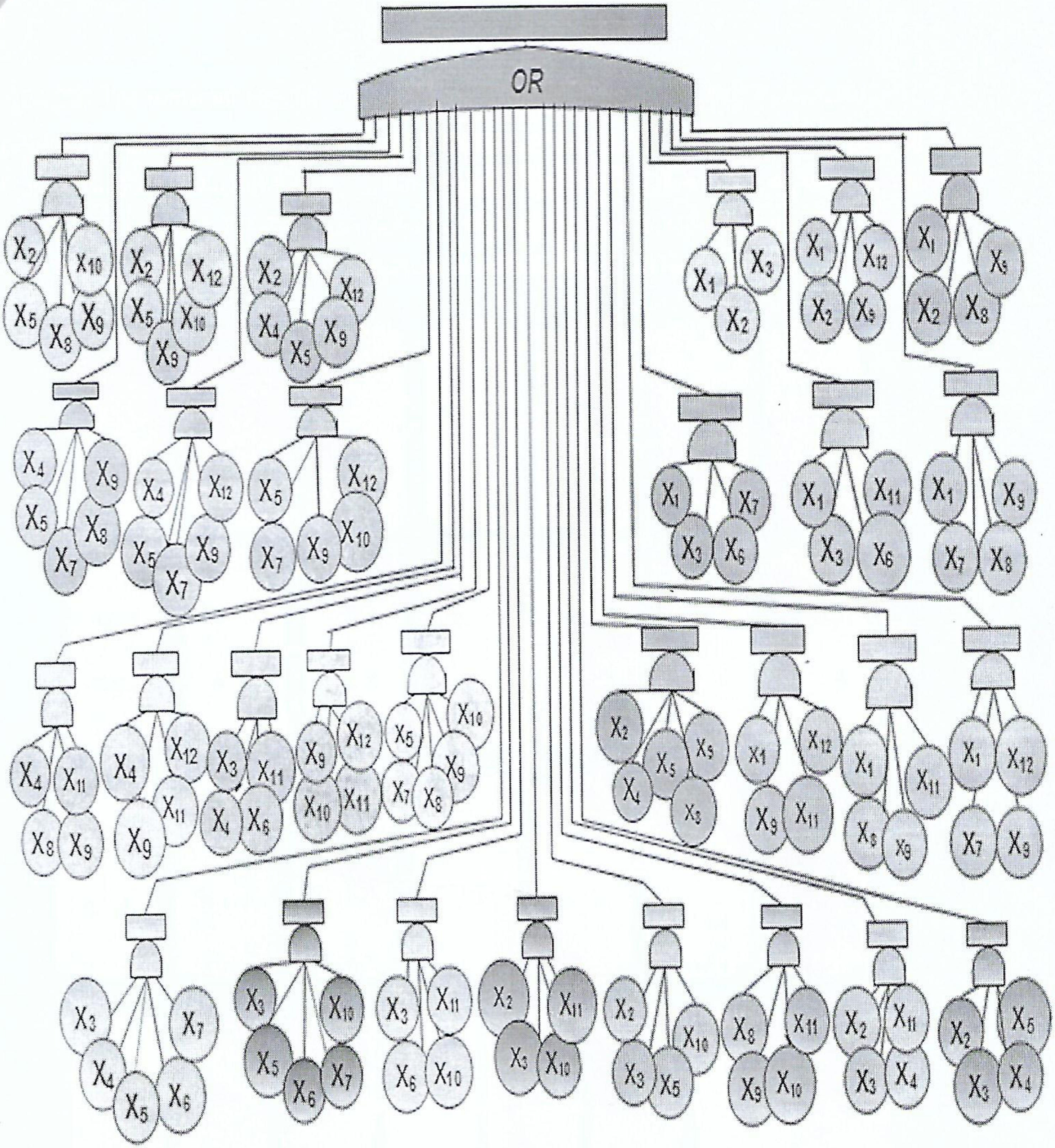



**Figure4:** A fault tree for the minimal cut sets of **Figure3(xv).**

[21,22] The minimal cut sets are $\{X_1,X_2,X_3\}$, $\{X_1,X_2,X_8,X_9\}$, $\{X_1,X_2,X_9,X_2\}$, $\{X_1,X_3,X_6,X_7\}$, $\{X_1,X_3,X_6,X_{11}\}$, $\{X_1,X_7,X_8,X_9\}$, $\{X_1,X_7,X_9,X_{12}\}$, $\{X_1X_8X_9X_{11}\}$, $\{X_1,X_9,X_{11},X_{12}\}$, $\{X_2,X_3,X_4,X_5\}$, $\{X_2,X_3,X_4,X_{11}\}$, $\{X_2,X_3,X_5,X_{10}\}$, $\{X_2,X_3,X_{10},X_{11}\}$, $\{X_3, X_4, X_6, X_{11}\}$, $\{X_3,X_6,X_{10},X_{11}\}$, $\{X_4,X_8,X_9,X_{11}\}$, $\{X_4,X_9,X_{11},X_{12}\}$, $\{X_8,X_9,X_{10},X_{11}\}$, $\{X_9,X_{10},X_{11},X_{12}\}$, $\{X_2X_4X_5X_8X_9\}$, $\{X_2,X_4,X_5,X_9,X_{12}\}$, $\{X_2,X_5,X_8,X_9,X_{10}\}$, $\{X_2,X_5,X_9,X_{10},X_{12}\}$, $\{X_3,X_4,X_5,X_6,X_7\}$, $\{X_3,X_5,X_6,X_7,X_{10}\}$, $\{X_4,X_5,X_7,X_8,X_9\}$, $\{X_4,X_5,X_7,X_9,X_{12}\}$, $\{X_4,X_5,X_8,X_9,X_{11}\}$, $\{X_5,X_7,X_8,X_9,X_{10}\}$

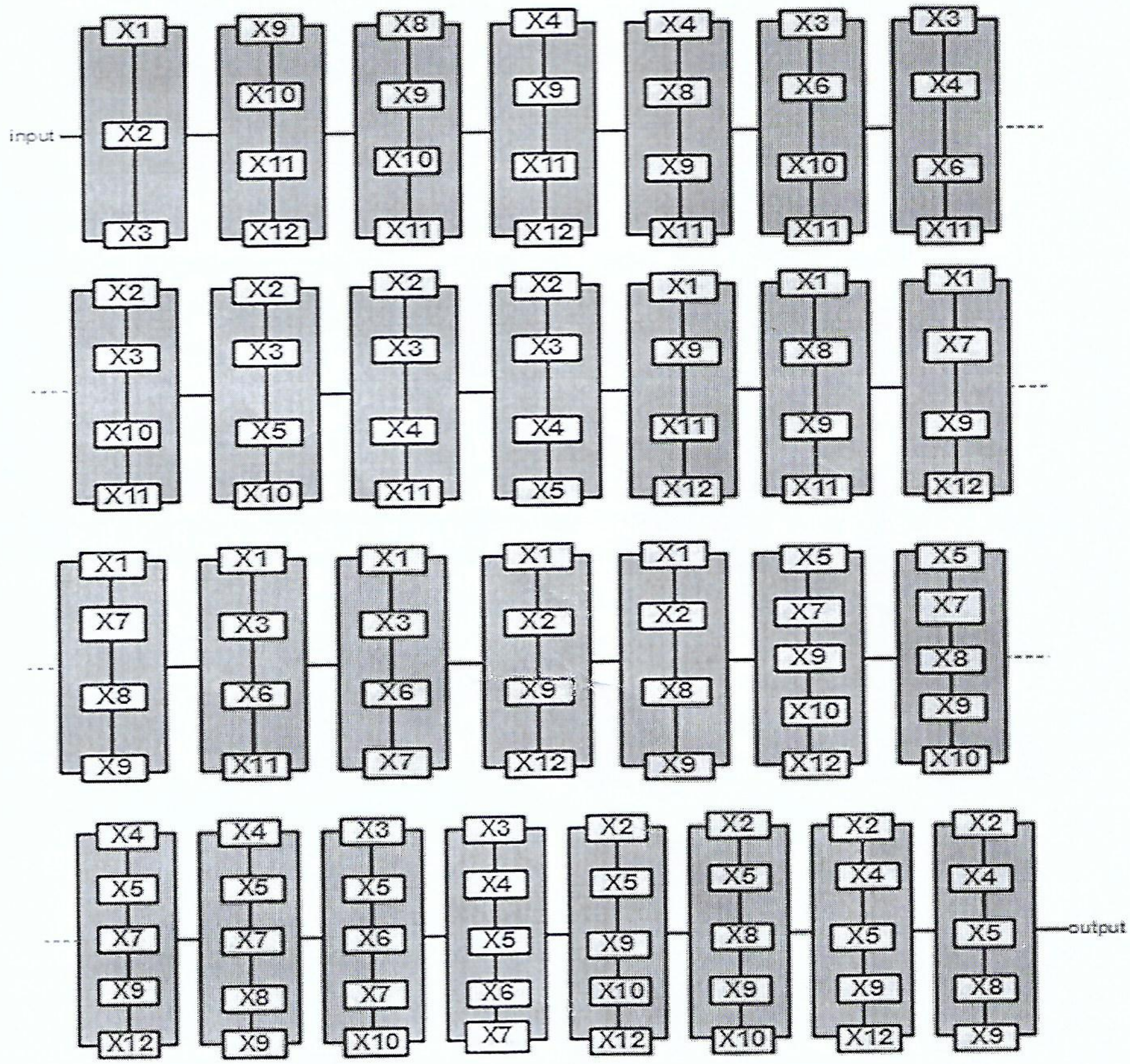



**Figure5:** Series–parallel system structure

The Reliability of a series - parallel system is

$$R_s=\prod_{i=1}^{m}(1-\prod_{j=1}^{n}(1-P_j)) \qquad (1)$$

$$\begin{aligned}
R_s=&R_3R_9+R_1R_4R_{10}+R_1R_5R_{11}+R_2R_6R_9+R_2R_7R_{11}+R_3R_8R_{12}-R_2R_3R_6R_9+R_2R_6R_8\\
&R_{12}-R_3R_8R_9R_{12}-R_1R_2R_5R_7R_{11}-R_1R_3R_4R_9R_{10}-R_1R_3R_5R_9R_{11}-R_1R_4R_5R_{10}R_{11}-\\
&R_2R_3R_6R_8R_{12}-R_2R_3R_7R_9R_{11}-R_2R_6R_7R_9R_{11}-R_2R_6R_8R_9R_{12}-R_1R_2R_4R_6R_9R_{10}-\\
&R_1R_2R_5R_6R_9R_{11}-R_1R_2R_4R_7R_{10}R_{11}-R_1R_3R_4R_8R_{10}R_{12}+R_2R_3R_6R_7R_9R_{11}-R_1R_3\\
&R_5R_8R_{11}R_{12}+R_2R_3R_6R_8R_9R_{12}-R_2R_3R_7R_8R_{11}R_{12}-R_2R_6R_7R_8R_{11}R_{12}+R_1R_2R_3\\
&R_4R_6R_9R_{10}+R_1R_2R_3R_5R_6R_9R_{11}+R_1R_2R_3R_5R_7R_9R_{11}+R_1R_2R_4R_5R_7R_{10}R_{11}+\\
&R_1R_2R_5R_6R_7R_9R_{11}-R_1R_2R_4R_6R_8R_{10}R_{12}+R_1R_3R_4R_5R_9R_{10}R_{11}-R_1R_2R_5R_6R_8\\
&R_{11}R_{12}+R_1R_3R_4R_8R_9R_{10}R_{12}+R_1R_3R_5R_8R_9R_{11}R_{12}+R_2R_3R_6R_7R_8R_{11}R_{12}+R_2\\
&R_3R_7R_8R_9R_{11}R_{12}+R_2R_6R_7R_8R_9R_{11}R_{12}-R_1R_2R_3R_5R_6R_7R_9R_{11}+R_1R_2R_3R_4R_6\\
&R_8R_{10}R_{12}+R_1R_2R_3R_4R_7R_9R_{10}R_{11}+R_1R_2R_3R_5R_6R_8R_{11}R_{12}+R_1R_2R_4R_5R_6R_9\\
&R_{10}R_{11}+R_1R_2R_3R_5R_7R_8R_{11}R_{12}+R_1R_2R_4R_6R_7R_9R_{10}R_{11}+R_1R_2R_4R_6R_8R_9R_{10}\\
&R_{12}+R_1R_2R_5R_6R_7R_8R_{11}R_{12}+R_1R_2R_5R_6R_8R_9R_{11}R_{12}+R_1R_3R_4R_5R_8R_{10}R_{11}R_{12}-\\
&R_2R_3R_6R_7R_8R_9R_{11}R_{12}-R_1R_2R_3R_4R_5R_6R_9R_{10}R_{11}-R_1R_2R_3R_4R_5R_7R_9R_{10}R_{11}-\\
&R_1R_2R_3R_4R_6R_7R_9R_{10}R_{11}-R_1R_2R_3R_4R_6R_8R_9R_{10}R_{12}-R_1R_2R_3R_5R_6R_7R_8R_{11}R_{12}\\
&-R_1R_2R_4R_5R_6R_7R_9R_{10}R_{11}-R_1R_2R_3R_5R_6R_8R_9R_{11}R_{12}+R_1R_2R_3R_4R_7R_8R_{10}R_{11}\\
&R_{12}-R_1R_2R_3R_5R_7R_8R_9R_{11}R_{12}+R_1R_2R_4R_5R_6R_8R_{10}R_{11}R_{12}+R_1R_2R_4R_6R_7R_8R_{10}\\
&R_{11}R_{12}-R_1R_2R_5R_6R_7R_8R_9R_{11}R_{12}-R_1R_3R_4R_5R_8R_9R_{10}R_{11}R_{12}+R_1R_2R_3R_4R_5R_6\\
&R_7R_9R_{10}R_{11}-R_1R_2R_3R_4R_5R_6R_8R_{10}R_{11}R_{12}-R_1R_2R_3R_4R_5R_7R_8R_{10}R_{11}R_{12}-R_1R_2\\
&R_3R_4R_6R_7R_8R_{10}R_{11}R_{12}+R_1R_2R_3R_5R_6R_7R_8R_9R_{11}R_{12}-R_1R_2R_4R_5R_6R_7R_8R_{10}\\
&R_{11}R_{12}-R_1R_2R_3R_4R_7R_8R_9R_{10}R_{11}R_{12}-R_1R_2R_4R_5R_6R_8R_9R_{10}R_{11}R_{12}-R_1R_2R_4R_6\\
&R_7R_8R_9R_{10}R_{11}R_{12}+R_1R_2R_3R_4R_5R_6R_7R_8R_{10}R_{11}R_{12}+R_1R_2R_3R_4R_5R_6R_8R_9R_{10}\\
&R_{11}R_{12}+R_1R_2R_3R_4R_5R_7R_8R_9R_{10}R_{11}R_{12}+R_1R_2R_3R_4R_6R_7R_8R_9R_{10}R_{11}R_{12}+R_1\\
&R_2R_4R_5R_6R_7R_8R_9R_{10}R_{11}R_{12}-R_1R_2R_3R_4R_5R_6R_7R_8R_9R_{10}R_{11}R_{12} \qquad (2)
\end{aligned}$$

If $(R_1,R_2,\ldots,R_{12})$ are independend identical, get the reliability polynomial.

$$R_s=R^2+5R^3-R^4-8R^5-5R^6+9R^7+8R^8-7R^9-5R^{10}+5R_{11}-R^{12} \qquad (3)$$

## IV.Conclusions

Many systems such as computer network, Nuclear power, electric power distribution, transportation, etc. can be modeled as networks or into fault trees first, so engineers can validate and verify their designs and evaluate system performance. Reliability is normally selected as one of the most important indices in many real world systems. Most of network reliability evaluation methods are formulated in terms of MCs. However, both the problems in locating all MCs and computing the network reliability in terms of the known MCs are NP-hard. This study presented another way is A very easy and more efficient MOCUS algorithm for finding all MCs between the source vertex and the sink vertex with time a complexity was developed.

## Reference


[1] C.Tanguy, Exact two-terminal reliability of some direct networks, arxiv:0807.0629v1[CS.PF] 3 Jul 2008.

[2] Y.-K. LIN AND P.-C. CHANG, Maintenance Reliability Of A Computer Network With Nodes Failure In The Cloud Computing Environment, *International Journal of Innovative Computing, Information and Control,* Volume **8**, Number **6**, June 2012, pp. 4045-4058

[3 ] Hassan, Z.A.H. and Mutar,E.K., Geomerty of reliability models of electrical system used inside spacecraft, Al-Sadiq Second International Conference on Multidisciplinary in IT and Communication Science and Applications 2017,IEEE,to appear.

[4] Sanjay Kumar Tyagi , Diwakar Pandey , Vinesh Kumar,2011,Fuzzy Fault Tree Analysis for Fault Diagnosis of Cannula Fault in Power Transformer.Scientific Research Corporation.

[5] Marco Bozzano , Adolfo Villafiorita,2003, Integrating Fault Tree Analysis with Event Ordering Information,ITC - IRST, Via Sommarive 18, Povo, 38050 Trento, Italy

[6] Jianwen X, Kazuhiro O ,Weiqiang K and Kokichi F,2006,From Fault Tree Analysis to Formal System Specification and Verification with OTS/CafeOBJ, Japan Society for Software Science and Technology, Vol.2, No.2, pp.448-460.

[7] Han Suk Pan &WonYoungYun,1997, Fault tree analysis with fuzzy gate Computers & Industrial Engineering Volume 33, Issues 3–4, December 1997, Pages 569-572.

[8] Sinnamon R M, 1996, Binary Decision Diagram, PhD Thesis, Loughborough University, UK. 31

[9] Dan M. S & Joseph T, 2007, Condition-based fault tree analysis (CBTA): A new method for improved fault tree analysis (FTA), reliability and safety calculations, Reliability Engineering and System Safety.

[10] Lee, W S, Grosz, D L., Tillman F A, Lie, C H, 1985, Fault Tree Analysis, Methods, and Applications – A review: IEEE Transactions on Reliability. 29.

[11] Jianwen X, Kazuo Y, Yoshiharu M, Kumiko T, Fumio M, Atsushi K and Takao O, 2011, Efficient Analysis of Fault Trees with Voting Gates.IEEE International Symposium on software Reliability Engineering.

[12] Dhillon, B. S., Design reliability: fundamentals and applications, CRC press, 1999.

[13] Limnios N. & Ziani R, 1986, Algorithm for Reducing Cut Sets in Fault Tree Analysis, IEEE Transactions on Reliability, Vol. 5.

[14] Ladislav R, 1996, Algorithm for finding minimal cut sets in a fault tree, Reliability Engineering and System Safety: Elsevier Science Limited.

[15] Antoine R, 1993, New Algorithms for fault trees analysis, Reliability Engineering and System Safety.

[16] Fault Tree Handbook with Aerospace Applications (updated NUREG-0492), 2002.

[17] Dhillon, B.S. and Singh, C., *Engineering Reliability: New Techniques and Applications,* John Wiley & Sons, New York, 1981.

[18] Barlow, R.E. and Proschan, F., *Statistical Theory of Reliability and Life Testing,* Holt, Rinehart and Winston, New York, 1975.

[19] Fussell, J.B. and Vesely, W.E., A new methodology for obtaining cut sets for fault trees, *Trans. Am. Nucl. Soc.,* 15, 262-263, 1972.

[20] Semanderes, S.N., Elraft, a computer program for efficient logic reduction analysisof fault trees, *IEEE Trans. Nuclear Sci.,* 18, 310-315, 1971.

[21] Kuo, W. and Zuo, M. J., Optimal reliability modeling: principles and applications, John Wiley and Sons, 2003.
[22] Yang, G., Life cycle reliability engineering, John Wiley and Sons,2007.