Republic of Iraq

Ministry of Higher Education

and Scientific Research

University of Babylon

College of Education for Pure Sciences

Mathematical Model of Reliability Optimization Techniques for Networks

A Dissertation

Submitted to The Council of The College of Education for Pure Sciences in University of Babylon in Partial Fulfillment of The Requirements for the Degree of Doctor of Philosophy in Education/Mathematics

By

Ghazi Abdullah Madlool Hedmah

Supervised by

Prof. Dr. Zahir Abdul Haddi Hassan

2021A.D 1443 A.H

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

وَيَسْأَلُونَكَ عَنِ الرُّوحِ ۖ قُلِ الرُّوحُ مِنْ
أَمْرِ رَبِّي وَمَا أُوتِيتُمْ مِنَ الْعِلْمِ إِلَّا قَلِيلًا 85

صدق الله العظيم

سورة الاسراء – الآية 85

Dedication

To my parents.

Who were the reason of what I become today.

Thanks for your great support and continuous care.

To my brothers, sisters, wife and my children

And to everyone who stand beside me to reach this success.

I am very grateful for your love and care.

Acknowledgements

The Name of Allah, the Most Merciful, the Most Compassionate all praise be to Allah, the Lord of the worlds; and prayers and peace be upon Mohamed, his servant and messenger.

First and foremost, I want to acknowledge my limitless thanks to Allah, the Ever-Magnificent for his help and bless.

My sincere thanks to my supervisor **Prof. Dr. Zahir Abdul Haddi Hassan** for his guidance, support and encouragement throughout the period of this work. I can't find the words to express my thanks and my gratitude to him.

Deep thanks go to **Dr. Hussein.J. Mutashar**, and **Hatem Kareem Sulaiman**, for their help and support.

My thanks are also extended to all members of the of the Mathematics Department (University of Babylon, College of Education for Pure Sciences).

Finally, I would like to express my deepest appreciation to my family and friends for supporting and encouraging me throughout my years of study and through the process of writing this thesis.

Supervisor's Certification

I certify that this dissertation **“Mathematical Model of Reliability Optimization Techniques for Networks”** by student "**Ghazi Abdullah Madlool"** was prepared under my supervision at the University of Babylon, Faculty of Education for Pure Sciences, in a partial fulfillment of the requirements for the degree of Doctor of Philosophy in Education / Mathematics.

Signature:

Name : Dr.

Title: Professor

Date: / / 2021

In view of the available recommendations, I forward this dissertation for debate by the examining committee.

Signature:

Name: Dr. Azal Jaafar Musa.

Head of Mathematics Department

Title: Assistant Professor

Date: / / 2021

Certification of linguistic Expert

I certify that I have read this dissertation entitled "**Some reliability optimization techniques for networks** " and corrected its grammatical mistakes; therefore, it has qualified for debate.

Signature:

Name: Dr.

Title:

Address:

Date: /   / 2021

Certification of Scientific Expert

I certify that I have read the scientific content of this dissertation "**Some reliability optimization techniques for networks**" and I have approved this dissertation is qualified for debate.

Signature:

Name: Dr.

Title:

Date:

Examining Committee Certification

We certify that we have read this dissertation " **Some reliability optimization techniques for networks** " and as committee, examined the student " **First second third fourth name** " in its contents and that in our opinion it meets the standard of a dissertation for the Degree of Doctor of Philosophy in Education / Mathematics.

Signature:
Name:
Title:
Date: / / 2021
Chairman

Signature:
Name:
Title:
Date: / / 2021
Member

Signature:
Name:
Title:
Date: / / 2021
Member

Signature:
Name:
Title:
Date: / / 2021
Member

Signature:
Name:
Title:
Date: / / 2021
Member /

Signature:
Name:
Title:
Date: / / 2021
Member / Supervisor

Approved by the Dean of the College.

Address: **Dean of the College of Education for Pure Sciences**.

Signature:

Name: Dr.Bahaa Hussein Salih Almuraab

Scientific grade: Professor

Date: / /2021

Contents

Chapter 4: Reliability Allocation and Importance

Chapter 5: Optimization Algorithms

Chapter 6: A Comparative Study Between GA, PSO,ACO and BCO to Evaluate the Reliability Optimization Problem for Complex System

Chapter 7: Conclusions and future works

List of Figures

List of Tables

List of Symbols

Symbol	Description
G	A graph defined by an ordered pair (V,E)
V	The set of vertices of G
E	The set of edges of G
$P_r(x)$	Probability of element x
$E[x]$	Expected value of element x
R_s	Reliability of the system
R_i	Reliability of the component i
CM	Connection matrix of the system
IM	Incidence matrix of all minimal paths
$ARINC$	Aeronautical Research Inc.
$AGREE$	Advisory Group of Reliability of Electronic Equipment
m_i	is the number of modules in subsystem i.
w_i	is the importance of subsystem i.
t_i	is the operating time of subsystem i.
m	is the total number of modules in the system
λ_S	is system failure rate.
T	Is mission duration.
C_k	Complexity of subsystem (k)
λ_k	Is failure rate allocated to each subsystem.
W	is sum of the rated product's.
R_U	Upper Bound on Reliability
R_L	Lower Bound on Reliability
$I_R(i)$	reliability importance

Symbol	Description
W_k	is rating for subsystem (k).
R_{ik}	is rating for each of the four factors for each subsystem.
$R_{i,min}$	Minimum reliability of component i
$R_{i,max}$	Maximum reliability of component i
R_G	System reliability goal
$C_i(R_i)$	Cost of component i
$C(R_1,..,R_n)$	Total system cost
R_j	Reliability of component j
A	Algorithm
GA	Genetic algorithm
PSO	Particle Swarm Optimization
ACO	Ant colony Optimization
BCO	Bees Colony Optimization

Abstract

This thesis aims to study of mathematical models in reliability of networks, as well as using some techniques due to calculate the reliability of complex network as an application of graph theory. Also, we will discuss important issues that include: using a mathematical method for creating minimal paths and minimal cuts for a network depending on the algebra of matrices, employ some methods to get the reliability polynomial for a network, including: path tracing, minimal cut method, reduction to series elements and the Inclusion - exclusion method. In addition to utilize some techniques to calculate the reliability allocation of shortened network such as Equal Allocation Technique, ARINC Approach method, AGREE allocation method and last Feasibility of objectives Technique. The thesis included calculating the importance of all components of the system to find the effect of all these components on the functioning of the system as a whole.

Finally, we will be calculating the reliability allocation and optimization for a complex network by using genetic algorithm, particle swarm optimization, ant colony algorithm and bee's colony optimization, with a comparison between these algorithms to choose the best algorithm that gives the highest reliability and lowest cost.

Chapter One

Introduction

1.1 Introduction

The phrase "reliability" dates back to 1816, when it was first used by poet Samuel Taylor Coleridge. Before World War II [69], the phrase was largely related with repeatability: a test (in any sort of research) was regarded dependable if the same results were produced repeatedly. The development of reliability theory has been strongly influenced by a series of accidents and catastrophic failures. Some of the achievements mentioned in this section may be difficult to comprehend fully at this stage. Established the theoretical foundation for using statistical methods in quality control of industrial products in the early 1930s, but such approaches were not widely applied until the outbreak of World War II. Despite the fact that they were comprised of individual high-quality components, products made up of a huge number of parts frequently failed. During his research on the strength of materials in the 1930s, the Swedish researcher Waloddi Weibull (1887–1979) made an important breakthrough. Weibull's (1939), [101]. He established the Weibull distribution, which is one of the most significant probability distributions in dependability theory (Weibull 1951), [100]. Lusser's law says that the reliability of series system is equal to the product of the reliabilities of individual components. Lusser was an engineer and aircraft designer who worked on Messerschmitt and Heinkel designs. An important contribution to the subsequent reliability theory was made by Russian mathematician Boris B. V. Gnedenko, [36]. In 1945, Milton A. Miner formulated the important Miner's rule for fatigue failures. In 1949, the Institute of Electrical and Electronic Engineers formed a professional group on quality control. The first guideline on failure modes and effects analysis was issued in 1949 (MIL-P-1629 1949)[73].

The Advisory Group on Reliability of Electronic Equipment (AGREE) was established in 1950 to survey the field [10]. The 1950s saw much pioneering work in the reliability discipline. The UK Atomic Energy Authority (UKAEA) was formed in 1954 and performed safety and reliability assessments for outside bodies. In 1960, the first edition of the US military handbook MIL-HDBK-217F [72] was published. In 1962, Bell Telephone Laboratories published a report on the safety of the Minuteman missile using fault tree analysis[22]. The Reliability Analysis Center (RAC) was established in 1968 for the US Department of Defense. Most important event for reliability in the 1970s was release of the Reactor Safety Study in 1975 (NUREG-75/014)[78]. Analysis of reliability and lifetime data grew more important. A high number of safety and reliability projects were sponsored by the oil and gas industry. 1980s started with a new journal Reliability Engineering, which had a great influence on the further development of reliability theory. Fault tree analysis got more standardized through the US NRC's Fault Tree Handbook. Bayesian probability entered into the field of reliability promoted by Martz and Waller.

After 1990, a range of new journals, books, education programs, computer programs and conferences emerged on the topic of system reliability. After 1990, the industry started to integrate reliability in their system development processes. The first edition of IEC 61508 "Functional safety of electrical/ electronic/programmable electronic safety-related systems" came in 1997,[47]. Since then, more and more software has been introduced in almost all types of systems. Software quality and reliability are now an important part of most system reliability assessments. The ability of a system to fulfill the task for which it is accountable at a specific moment is described by reliability theory. It is one of the engineering cornerstones. This helps to increase the performance of systems and lessen the likelihood of failure, such as aircraft, linear accelerators, and other products. The growth of reliability has taken on a new dimension in recent years, owing to the consequences of failures of today's complex systems, which can cause daily annoyance to operational exigency and uneconomical maintenance, and even to the point of endangering life on our planet if quality and reliability are compromised. disastrous. Complex networks are becoming increasingly important in a range of fields, including biology, engineering, and the social sciences.

In network science, graph theory theories are utilized to model systems of interest. Network research is primarily concerned with the deployment of dynamic processes across networks. For global epidemics, reported operations include energy, information, or commodity exchange of chemical atoms. We examine these diusive processes on networks and the various research problems that have arisen around this topic using network reliability. In their 1950s investigation of the reliability of relay circuits, Shannon and Moore [106] addressed network reliability for the first time. They defined network reliability as the probability that a circuit will remain closed if all of the circuit's constituent contacts are closed with a specified probability. . Since their work was first published more than half a century ago, a great amount of research has centered on this topic. Per formability, survivability, and performance are all phrases that are used to describe the same thing in different fields. As a result, network reliability is a probabilistic metric that determines whether a network can continue to function even if one or more of its components fails at random. Because the definition of functionality varies depending on the situation at hand, network reliability offers a lot of potential as a unifying framework for studying a wide range of issues that arise in complex network environments. In network reliability analysis and design, there are basically two primary areas of research [82]. The purpose of analysis is to measure the dependability of a given network, whereas the goal of design is to provide design engineers with techniques to improve their capacity to design networks with a high level of reliability. In an ideal world, one would wish to create a network. In recent years, there has been a significant surge in interest in network resilience, particularly telecommunications network reliability [71]. Graph theory has grabbed the curiosity of scientists and engineers in recent decades. The demonstrated ability of graph theory to address problems from a wide range of domains is the fundamental reason for its increased popularity. Graphs have been proven to be particularly beneficial in modeling systems originating in physical science, engineering, social sciences, and economic problems due to their straightforward diagrammatic form, and reliability engineering is no exception. The application of graph theory to reliability studies received little attention till [90]1970 .Following that, the literature has oered a number of methods, strategies، and approaches. The application of graph theory has become inextricably linked to the assessment of network reliability. reliability has become a key factor in the design and operation of modern large, complex, and expensive networks. System design means certain features (cost, reliability and weight) can be characterized as an issue of combinatorial optimization where either reliability can be improved or costs can be lowered. Technology development has brought changes to different fields of contemporary culture(such as mathematics, productivity and health). Reliability of the system has become increasingly crucial for all involved parties (designers, suppliers and users) in this context. During the design phase, the reliability of the product is evaluated. Increasing product reliability triggers cost increases automatically, which has a adverse effect on the consumer. As a consequence, the problem becomes how to satisfy the minimum price criteria for system reliability. An alternative in this context is to use redundant parts to improve the reliability of the system. System design means certain features (cost, reliability and weight) can be characterized as an issue of combinatorial optimization where either reliability can be improved or costs can be lowered. There are limitations on price, weight or system reliability targets in either of the circumstances. The issue with reliability redundancy distribution is to pick the highest mix of parts and redundancy levels to either maximize system reliability and minimize system costs subject to multiple limitations. There are three kinds of reliability optimization issue: reliability allocation (variables are system component reliability), redundancy allocation (variable is redundant component number) and reliability redundancy allocation (variables are system component reliability and redundant component number).

1.2 Historical Review of Reliability Optimization Problems

Our society is now largely reliant on current technology systems, which have unquestionably benefited our society's productivity, health, and wealth. However, as people become more reliant on modern technology systems, they must deal with increasingly complex operations and management[10]. The system reliability plays a critical part in each of the complex/complicated systems. Manufacturers, designers, and users all place a premium on system reliability. Reliability engineers/designers are hired during the product development process to assess the product's reliability. They want their products to be more reliable, which increases the cost of production. In this instance, the question of how to achieve the system reliability aim emerges. As a result, the rise in production costs has a negative impact on the budget of the user. As a result, the design reliability optimization challenge is defined as minimum reliability improvement cost. In this regard, introducing numerous redundant components is a well-known way for increasing a system's reliability. The corresponding problem can be framed as a combinatorial optimization problem, where either system dependability is maximized or system cost is minimized, for better building a system employing components with known cost, reliability, weight, and other qualities. As a result, both formulations typically include restrictions on the maximum permissible weight, cost, and/or minimum goal system dependability level. The reliability redundancy allocation problem is the comparable problem. In 1972, Kim, Y. H [57] gave the algorithm a list of possible routes from the source to the graph sink to calculate system reliability based on route data. A network is reliable when a route connects every pair of nodes. Reliability analysis of networks such as networks for computer architecture and data communication networks.In 2015 Hassan, Z. A. H. and Balan, V. [39], introduced the quadratic case of reliability, to make use the convex/concave mutual dependence of slice-components along the curves of constant-slice reliability, in order to maintain or increase the circuit reliability. Clear techniques of engineering to evaluate the reliability of electrical systems used in aircraft show that elements of reliability must be linearly based on time[42]. In 2017 Kumar, A., Pant, S.[60], Ram, M., Singh, S. B. also presented solving complex reliability optimization problem using multi-objective particle swarm optimization. In the same year, Abed, S. A. and Udriste, C., presented Optimization Techniques [5] and Methods in reliability allocation also, Shaghaghi, Saba, [91]. "Comparative analysis of GMDH neural network based on genetic algorithm and particle swarm optimization in stable channel design". In 2018 Feng, X., Zhu, X., Zhao, W., and Li, X. have introduced Reliability of Electric Vehicle with Wind Turbine Based on Particle Swarm Optimization [33]. In 2019 Ouyang, Z., Liu، Y., Ruan, S. J. and Jiang, T. have presented an improved particle swarm optimization algorithm for reliability redundancy allocation problem with mixed redundancy strategy and heterogeneous components [79]. Abbas Abed, S., Kareem Sulaiman, H., Abdul Haddi Hassan, Z. have presented Reliability Allocation and Optimization for(ROSS) of a Spacecraft by using Genetic Algorithm [7].

The aim of this thesis is to finding the best system optimization using four algorithms (genetic algorithm ,particle swarm optimization, ant colony algorithm and Bees Colony Optimization) and three cost functions (exponential behavior with feasibility factor model, exponential behavior model and logarithmic model) and comparing the results for the four algorithms of the three cost functions. This thesis consists of seven chapters, while the given order is the one suggested.

Chapter 1 Introduction
Chapter 2 Contains basic definitions and general concepts

Chapter 3 Consists of a number of techniques for analyzing the reliability of complex system simplify systems in a straightforward way. The calculating of reliability based on the logic diagram.

Chapter 4 We are discussing the methods of solving the problem of allocating reliability to complex system after reducing the system in three components in parallel that became a subsystem . In addition, these methods used to solve the problem of allocation have been modified from series system to parallel system. The purpose of reliability allocation is to establish an objective for the reliability of each component, so that the manufactures can have an idea of the performance required of this product.

Chapter 5 In this chapter, we show the four algorithms (genetic algorithm, particle swarm optimization, ant colony algorithm, bee colony optimization), their schemes and working methods.

Chapter 6 Calculates the allocation and optimization of reliability to the complex system using the genetic algorithm and the particle swarm optimization, , ant colony algorithm, bee colony optimization with three costs (exponential behavior with feasibility factor model, exponential behavior model and logarithmic model)and compare the results.

Chapter 7 Consists of conclusions and future work.

Note: Most of the calculations made in this thesis have been done in Matlab program

Chapter Two

Basic Definitions and Concepts

2.1 Introduction

This chapter dealt with two main parts of the thesis, namely the basics of graph theory and the basics of reliability.

2.2 Basics of Graph Theory

In this part [8,27], we have covered various definitions, basic topics, types of direct and undirect graphs, simple and Multigraphs, Complete graphs, partial graphs, etc., and matrix representations of graphs[59,104].

Definition 2.1[100] *A* **graph** G is a set V of vertices together with a set E of edges. It is written as

G = (V,E).

Definition 2.2 [20] The order of the graph is a number of vertices in a given graph**,** denoted by $|V(G)|$.

Definition 2.3 [76] The number of edges in a given graph is called *size* of the graph, denotedby $|E(G)|$.

Definition 2.4 [103] All edge of a graph that joins a vertex to itself is called a *loop***.**

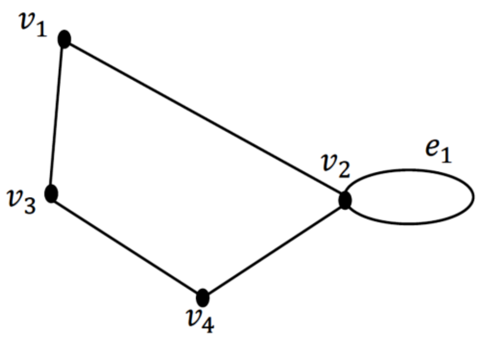


Figure 2.1: Graph with loop

Definition 2 .5 [26] The multiple edges if two vertices of a graph are joined by more than one edge. See Figure 2.2 (e_1 and e_2).

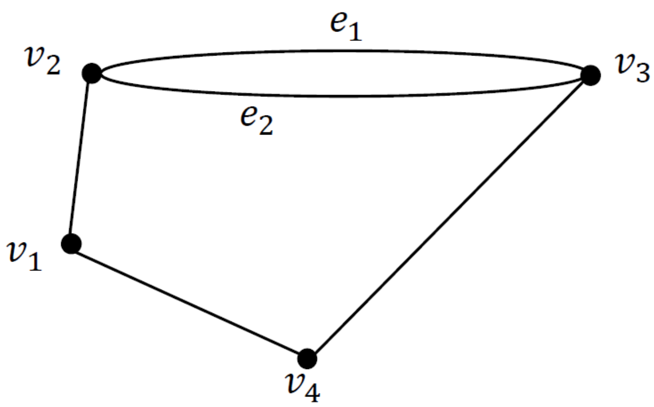


Figure 2.2: Graph with multiple edges

Definition 2.6[103] Simple Graphs and Multigraphs. A simple graph G is one that does not contain loops and parallel edges. A multigraph is a graph that is more complicated than simple Graph.

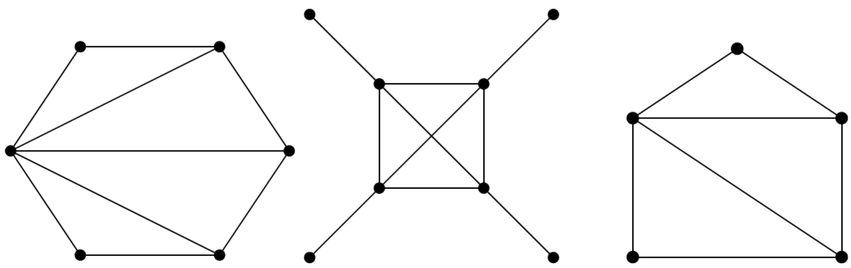


Figure 2.3: Some examples of simple graphs

Definition 2.7 [19] Graphs, both finite and infinite. A finite graph is one that has both a finite number of vertices and a finite number of edges. Otherwise, it's a never-ending graph..

Definition 2.8 : [44]Adjacent vertices are two vertices of a graph connected by an edge.

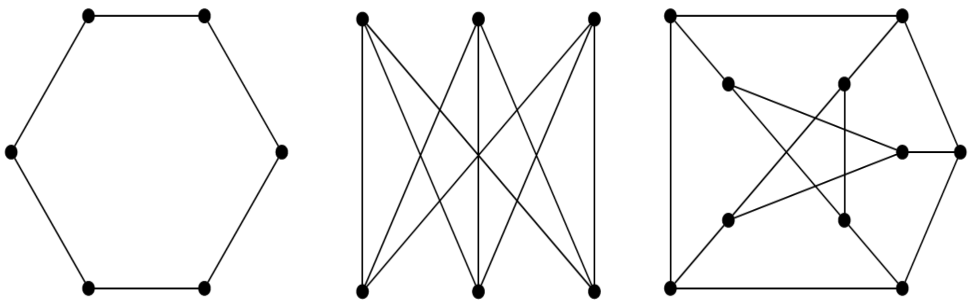
Definition 2.9 [27]Adjacent edges are defined as two or more edges in a graph that share a common vertex.

Definition 2.10 [75]Any graph's degree of a vertex v. The number of edges incident on v is defined by G. It's written as deg(v) or d(v).

Definition 2.11[19] An isolated vertex has a degree of 0 whereas an end-vertex, leaf, or pendant vertex has a degree of 1.

Definition 2.12 [100] A graph H is a *subgraph* of G if every vertex of H is a vertex of G, and every edges of H is an edge of G. In other words, $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.

Definition 2.13[103] Regular Graphs. If all of the vertices of a graph G have the same degree, the graph is said to be regular. If $d(v) = k \ \forall \ v \in V(G)$, a graph G is said to be a k-regular graph (G). An $(n - 1)$-regular graph is a complete graph.



(a)A 2-regular graph **(b) A 3-regular graph** **(c) Petersen Graph**

Figure 24: Examples of regular graphs

Definition 2.14 [13] A path is a set of components P such that if all of them are successfully operating then the system is operate,

Definition 2.15 A path set is minimal path set [94] if it cannot be reduced without losing its status as a path set.

Definition 2.16 [27] A cut is a set of components C such that if all of them are fail then the system is fail.

Definition 2.17 A cut set is minimal cut set [13] if it cannot be reduced without losing its status as a cut set .

Definition 2.18 **[76]** Connectedness in a Graph. If there is a path between two vertices u and v, they are said to be linked. If a path exists between two vertices u and v, then u can be reached from v and vice versa. It is said that a graph G is connected.

Definition 2.19 [76] Cut-Edge. A graph's edge (e) If G e is unconnected, G is said to be a cut-edge or a bridge of G..

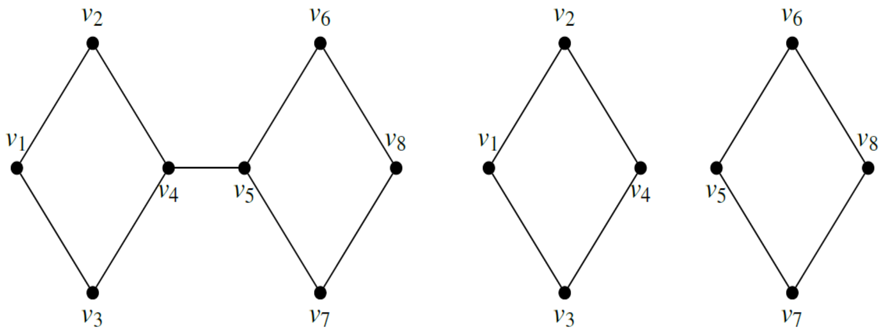


Figure 2.5: Disconnected graph $G-v_4v_5$

Definition 2.20 [76] Cut-Vertex. If G−v is unconnected, a vertex v of a graph G is said to be a cut-vertex of G.

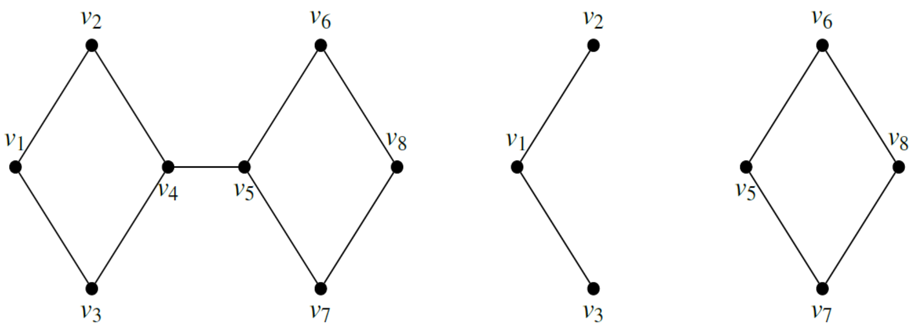


Figure 2.6: disconnected graph $G-v_4$

Definition 2. 21[100] Directed Graphs. A directed graph, also known as a digraph, is made up of a set V of vertices and a set E of edges, each of which is linked to an ordered pair of vertices. In other terms, a directed graph is one in which each edge of the graph G has a direction.

Definition 2.22 [76] Sources and Sinks. A source is a vertex with zero in-degree, whereas a sink is a vertex with zero out-degree.

Definition 2.23[103] If all edges of the graph **G** are undirected edges, then **G** is called an **undirected graph.**

Definition 2.24 [76] A graph **G** with both directed and undirected edges is called a **mixed graph.**

2.3 Matrix Representations of Graphs

Matrices are a different means of representing and analyzing network data. A matrix is similar to a graph in that it contains the same information, but it is more useful for computing and computer analysis. Indeed, there are numerous matrices associated with a properly labeled graph [76,103].

2.3.1 Incidence Matrix of a Graph

Without self-loops. The incidence matrix A of G is an $n \times m$ matrix defined by $A(G) = [a_{ij}]$; $1 \le i \le n$, $1 \le j \le m$. Let G be a graph with n vertices, m edges and

$$a_{ij} = \begin{cases} 1; & \text{if j} - \text{th edge incidents on i} - \text{th vertex;} \\ 0; & \text{otherwise..} \end{cases}$$

where the n rows of A in G correspond to the n vertices and the m columns of A to the edges[76,100] . There are just two sorts of elements in the incidence matrix: 0 and 1. As a result, this is unmistakably a binary matrix or a (0, 1) matrix..

The following table gives the incidence relation of the graph G:

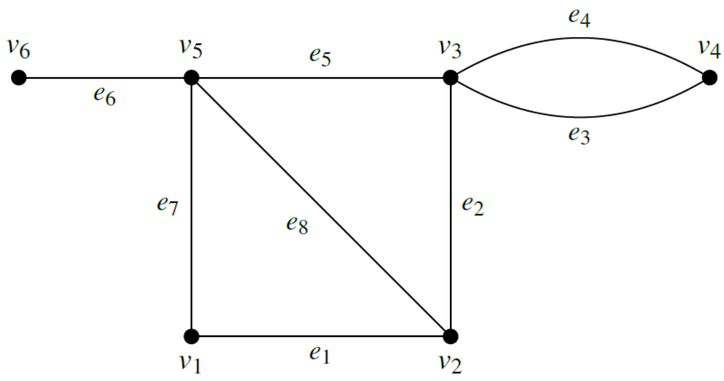


Figure 2.7: A graph *G*

Table 2.1: The incidence relation of the graph G

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
v_1	1	0	0	0	0	0	1	0
v_2	1	1	0	0	0	0	0	1
v_3	0	1	1	1	1	0	0	0
v_4	0	0	1	1	0	0	0	0
v_5	0	0	0	0	1	1	1	1
v_6	0	0	0	0	0	1	0	0

As a result, the G incidence matrix is as follows::

$$A(G) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Now consider the following disconnected graph *G* with two components.

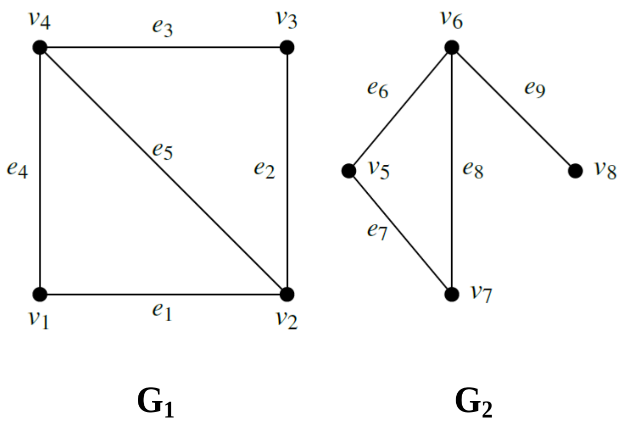


Figure 2.8: A disconnected graph *G* with two components G_1 and G_2.

The incidence matrix of the graph in Figure 2.8 is given below:

$$A(G) = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

2.3.2 Adjacency Matrix

As an alternative to the incidence matrix, it is sometimes more convenient to represent a graph by its adjacency matrix or connection matrix. The adjacency matrix of a graph G with n vertices and no parallel edges is an n by n symmetric binary matrix $X = [x_{ij}]$ defined over the ring of integers such that[76].

$$x_{ij} = \begin{cases} 1; & \text{if there is an edge between } i-\text{th and } j-\text{th vertices, and} \\ 0; & \text{if there is no edge between them.} \end{cases}$$

Consider the following graph G without parallel edges and self-loops.

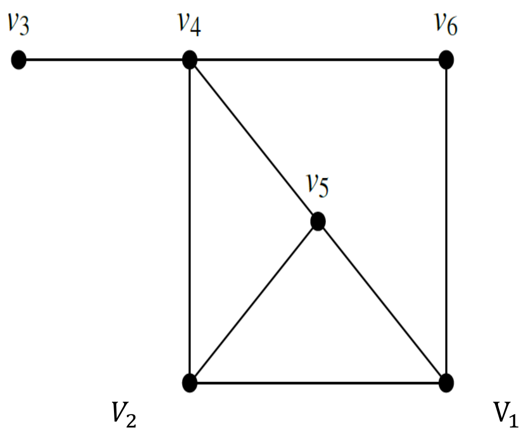


Figure 2.9 *G* graph

The adjacency matrix of the G graph in Figure 2.9 is given by

$$X(G) = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

The adjacency matrix of the disconnected graph given in Figure 2.8 is as follows:

$$X(G) = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

2.4 Basics of the Reliability Polynomials

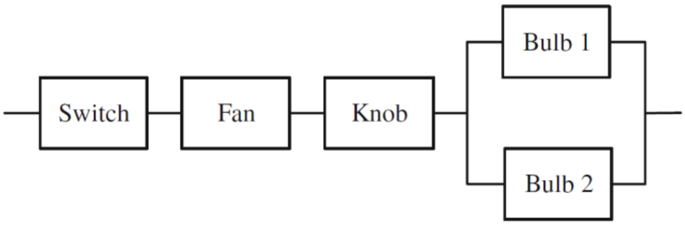
In this part, the main topics in reliability are discussed[57,75], including some definition as well Reliability block diagram, structure functions, reliability systems.

Definition2.25 [96]The multiaffine polynomial RG(p) is called multivariate reliability poly- nomial for graph G.

Definition 2.26[43]The probability that the system survives from some a specified period oftime is called the **reliability function**.

2.5 Reliability Block Diagram

The relationship between the functioning of a system and the functioning of its components is frequently depicted using a dependability block diagram. Take, for example, an overhead projector[18]. Assume that the system contains two light bulbs. Figure 2.10 shows the overhead projector's dependability block diagram. A rectangle or a circle is frequently used to represent a component in a reliability block diagram. The component's name can be specified in the block. Most of the time, the block just contains the label of each component. The five primary components of the system are depicted in Figure 2.10's reliability block diagram. It also signifies that the system is operational if and only if the switch, fan, knob, and at least one of the two bulbs are operational. A reliability block diagram may not always depict how the system's components are physically connected. It simply explains how the components' proper operation ensures the system's proper operation. As a result, a reliability block diagram depicts the logic relationship between the system's operation and that of its components[56]. A reliability block diagram, as shown in Figure 2.10, is best understood when the signal flows from left to right through the components. The signal can pass through a working component, but not through a failed one. Series structures, parallel structures, series–parallel structures, parallel–series structures, bridge structures, and generic network topologies have all been represented using reliability block diagrams. When these structures are introduced, diagrams will be provided. We typically use the number n to signify the number of components in a system, and each component is given a unique name from the set 1, 2,..., n in talks of system structures. C stands for the collection of components of a system



.

Figure 2.10 Reliability block diagram for overhead projector

2.6 Structure Functions

We assume that the system and its component can only be in one of two states: working or failed, for the majority of this thesis. As a result, each component's or system's state is a discrete random variable with just two potential values, [43] signifying the working and failure states, respectively. Let x_i represent the current state of component i, for $1 \le i \le n$ and

$$x_i = \begin{cases} 1; & \text{if component i works,} \\ 0; & \text{if component i is failed.} \end{cases}$$

Then, vector $x = (x_1, x_2, \ldots, x_n)$. The component state vector, which represents the states of all components, is known as the component state vector. The state of the system is also a binary random variable that is entirely determined by the components' states[57,75]. Let φ denote the current state of the system, and

$$\varphi = \begin{cases} 1; & \text{if the system works,} \\ 0; & \text{if the system is failed} \end{cases}$$

The system state is known if the states of all components are known. The system's state is a deterministic function of the components' states. As a result, we frequently write

$$\varphi = \varphi(x) = \varphi(x_1, x_2, \ldots, x_n)$$

and the system's structure function is referred to as φ(x). Each distinct system structure has a distinct structure function φ (x). To denote a system with the set of components C and the structure function, we typically use (φ, C).

Definition 2.27 [95] If the state of the system is not affected by the state of this component, it is irrelevant to the system's performance. In terms of mathematics, component i ($1 \leq i \leq n$) If and only if $(1_i, x) = (0_i, x)$ for each component state vector x, is irrelevant to the structural function. The component is said to be relevant if it is not otherwise.

Definition 2.28 [18] φ(x) is nondecreasing in each parameter x_i for $1 \leq i \leq n$ and every component is relevant If and only if a system with structure function φ(x) is coherent..

Definition 2.29 [18] The **structure function** φ of a network with n edges is a mapping $\varphi:\{0,1\}^{\mathbf{n}} \rightarrow \{0,1\}$, n is called order of the network.

2.7 Reliability Systems

A device or system is described as a collection of parts or components. The system operates successfully if all its components operate successfully (do not fail), but it may also operate if a subset of components has failed. The structure function is a model that determines the status of the system given the status of its components .

2.7.1 Simple Reliability Systems

Suppose that we have to calculate the reliability of a system made up of several components. The total reliability can be achieved, by calculating the reliability of each individual component, and combining these individual reliabilities depending on the ways they are connected[75]. That is, whether they are connected as series, parallel, series-parallel, parallel-series

2.7.1.1 Series Systems

If the failure of one or more components within a system causes the entire system to fail, the system is said to be a series system[13]. To put it another way, for a system to succeed, all of its components must work together. A general series system's reliability can be calculated as follows. Assume that a series system consists of n components that are mutually independent. The term "mutual independence" refers to the fact that the failure of one component has no bearing on the life of the others. The following notation is used: E_i indicates that component I is operational, E indicates that the system is operational, and R_i indicates component i's reliability. R stands for the system's dependability[43]. By definition, a system's successful operation necessitates the operation of all of its components[58]. The system's reliability can be calculated using probability theory.

$$R = Pr(E) = Pr(E_1 \cdot E_2 \ldots E_n).$$

Because of the premise of independence, this becomes

$$R = Pr(E_1)\, Pr(E_2)\ \cdots\ Pr(E_n) = \prod_{i=1}^{n} R_i \qquad (2.1)$$

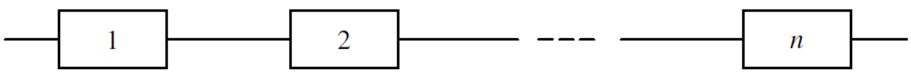


Figure 2.11 General series system

2.7.1.2 Parallel Systems

A system is said to be a parallel system if and only if the failure of all of its components causes the entire system to fail[58,75]. In other words, if one or more components are operational, a parallel system succeeds. A lighting system with three lights in a room, for example, is a parallel system since room blackout occurs only when all three bulbs fail. Figure 2.12 depicts the lighting system's dependability block diagram. The following formula is used to calculate the reliability of a general parallel system. Assume that a parallel system consists of n components that are mutually independent[11,43]. The following notation is used: The event that component I is functioning is called E_i.

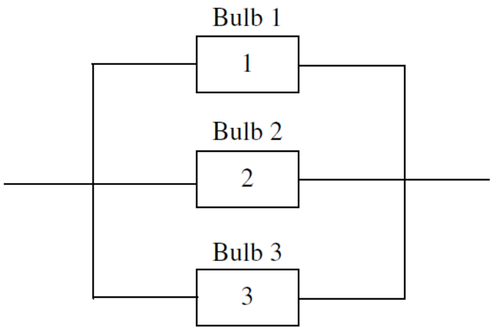


Figure 2.12 Reliability block diagram of the lighting system

E in the event that the system is up and running, $\bar{x}$ is of X complement, where X stands for E_i or E; Ri is the component's reliability; F is the system's unreliability (failure probability); and R is the system's reliability. In order for a parallel system to fail, all n components must fail. The unreliability of a system is defined by probability theory.

$$F = p_r\,(\overline{E}) = p_r\,(\overline{E_1} \cdot \overline{E_2} \cdot \cdot \cdot \overline{E_n}).$$

Because E_i (i = 1, 2, . . . , n) are mutually independent, we can write this equation as

$$F = p_r(\overline{E}_1)p_r(\overline{E}_2) \cdots \ p_r(\overline{E}_n) = \prod_{i=1}^{n}(1 - R_i). \qquad (2.2)$$

The complement of system unreliability is system reliability: thus,

$$R = 1 - \prod_{i=1}^{n}(1 - R_i). \qquad (2.3)$$

2.7.1.3 Mixed Compositions

In order to meet functional or reliability requirements, series and parallel topologies are sometimes blended in a system design[25,43]. Series–parallel and parallel–series configurations result from the combinations[15,24]. The reliability of these two types of systems is discussed in this section.

2.7.1.3.1 Series–Parallel Systems

As shown in Figure 2.13, a series–parallel system consists of n subsystems in series with mi (i = 1, 2,...,n) components in parallel in subsystem i. Low-level redundancy is a term used to describe this design.

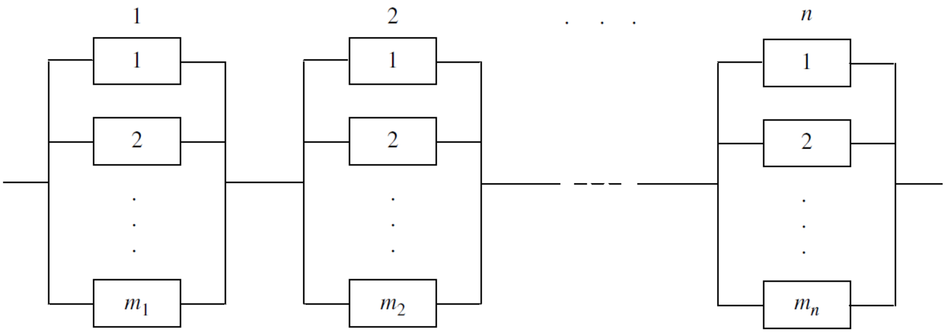


Figure 2.13 General series–parallel system

design. To determine the system's reliability, each parallel subsystem is first reduced to an equivalent reliability block. The block i reliability R_i is (2.3) , given by.

$$R_i = 1 - \prod_{j=1}^{m_i} (1 - R_{ij}). \tag{2.4}$$

where R_{ij} is the component j's reliability in subsystem i, (i= 1, 2,..., n) and (j = 1, 2,..., m_i). As seen in Figure 2.13, the n blocks form a series system that is equivalent to the original system. The system reliability R is then calculated.

$$R = \prod_{i=1}^{n} R_i \tag{2.5}$$

2.7.1.3.*2* Parallel–Series Systems

As shown in Figure 2.14, a generic parallel–series system consists of m subsystems running in parallel with n_i (i = 1, 2,...,m) components in subsystem i. High-level redundancy design is another name for this configuration. To calculate the system reliability, each series subsystem is first collapsed into an equivalent reliability block. The block I reliability Ri is calculated from (2.1).

$$R_i = \prod_{j=1}^{n_i} R_{ij} \qquad i = 1, 2, \dots, m \tag{2.6}$$

where R_{ij} denotes the component j's dependability in subsystem i. As demonstrated in Figure 2.14, the m blocks form a parallel system that is equivalent to the original. By substituting (2.6) for (2.3), the parallel–series system's dependability becomes

$$R = 1 - \prod_{i=1}^{m} (1 - R_i) \tag{2.7}$$

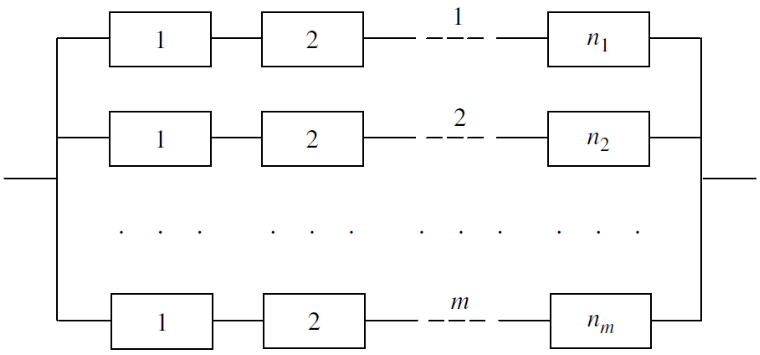


Figure 2.14 General parallel–series system

2.8. Complex System

In many circumstances, determining which components are connected in series and which are connected in parallel in a complicated system is difficult[43,75]. The network depicted below is an excellent illustration of a complex system

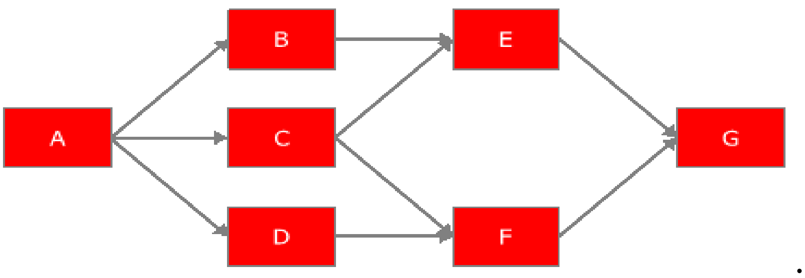


Figure 2.15: Complex system

The system depicted in the diagram above cannot be divided into series and parallel systems. This is owing to the fact that has two roads leading away from it, whereas only has one. There are several ways for determining a complicated system's reliability, including:

1. Path Tracing Method (Tie-Set Method).
2. Minimal Cut Method.
3. Reduction to Series Element Method.
4. The Inclusion-Exclusion Method.

Chapter Three

Methods to Evaluate the Reliability of Networks

3.1 Introduction

The techniques in Chapter two are only applicable to systems and networks with a series and parallel topology. Many systems either lack this basic structure or have complicated operational logic. In order to determine the reliability of such systems, further modeling and evaluation methodologies are required. The bridge is an example of a system that does not have a series/parallel structure. In this chapter, we discuss the a number of techniques for analyzing reliability to complex network system as shown in figure.3.1. In order to simplify the reliability of the system [28], these techniques are based on a logical diagram and the subject is addressed through some basic concepts in the graph theory as in the first chapter. A visual examination of the network in figure 3.1 reveals that none of the components are connected in a straightforward series/parallel configuration. Also, The following techniques were addressed : Connection matrix techniques , Node removal ,Matrix multiplication [43], Generation of minimal cut sets, Path Tracing Method (PTM), Minimal Cut Method, Reduction to Series Elements Inclusion–Exclusion Method, were used to obtain the reliability polynomial[51].

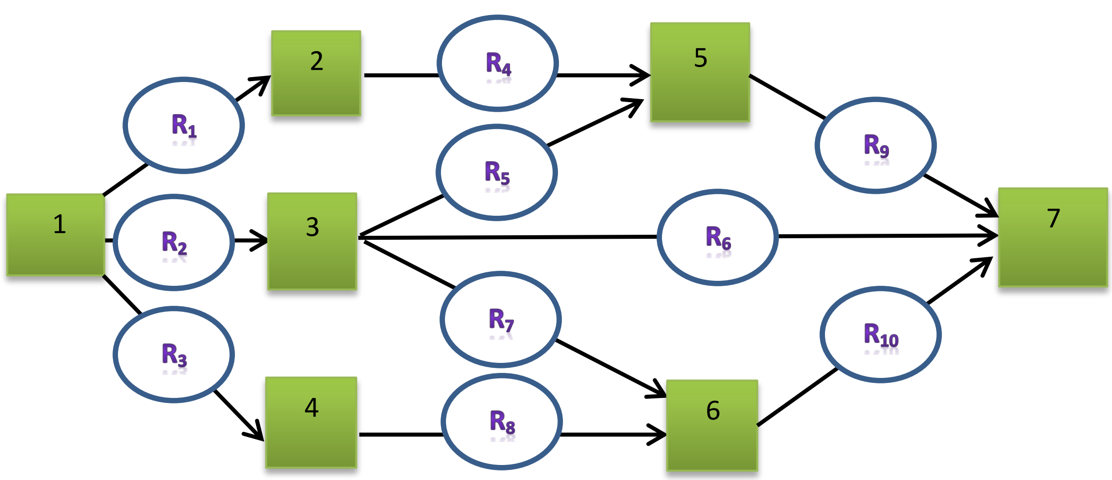


Figure 3. 1: Complex network

3.2 Connection Matrix Techniques

A connection matrix is created from the system network or reliability diagram in this manner, which defines which components are connected between the network or diagram's nodes. Reconsider figure (3.1) and identify the nodes as shown in figure (3.1), and construct the following connection matrix in which a zero indicates that there is no connection between two nodes and unity represents a connection between a node and itself, this being the value of the elements on the principal diagonal.

$$A = \begin{bmatrix} node & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 1 & x_1 & x_2 & x_3 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & x_4 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 & x_5 & x_7 & x_6 \\ 4 & 0 & 0 & 0 & 1 & 0 & x_8 & 0 \\ 5 & 0 & 0 & 0 & 0 & 1 & 0 & x_9 \\ 6 & 0 & 0 & 0 & 0 & 0 & 1 & x_{10} \\ 7 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

Unidirectional (flow just in one direction) and bidirectional (flow in both directions) branches are both included in this method[50]. The goal of this method of solution is to convert this fundamental connection matrix into one that describes the flow transmission between the input and output, or between the two nodes of interest. This can be accomplished in one of two ways: by removing nodes or by multiplying matrices.

3.2.1 Node Removal

By sequentially reducing the basic connection matrix [9]until it is reduced to a 2 x 2 matrix including only the input and output nodes, all nodes in the network that are not input or output are deleted[62]. In this case[43,75], the matrix must be simplified to only nodes 1 and 7. To delete node k from a matrix, each element must be removed. $N_{ij}(i,j \neq k)$ must be replaced according to

$$\acute{N}_{ij} = N_{ij} + (N_{ik}N_{kj}) \tag{3.2}$$

where $\acute{N}_{ij}$ replaces the old N_{ij} .

Consider the removal of node 2 first in this example.. We apply the equation (3.2) to the matrix (3.1) and we get.

The new matrix elements are

$N_{11} = 1 + x_1 \bullet 0 = 1$

$N_{13} = x_2 + x_1 \bullet 0 = x_2$

$N_{14} = x_3 + 0 = x_3$

$N_{15} = 0 + x_1 x_4 = x_1 x_4$

$N_{16} = 0 + x_1 \bullet 0 = 0$

$N_{17} = 0 + x_1 \bullet 0 = 0$

$N_{31} = 0 + 0 \bullet 0 = 0$

$N_{33} = 1 + 0 \bullet 0 = 1$

$N_{34} = 0 + 0 \bullet 0 = 0$

$N_{35} = x_5 + x_4 \bullet 0 = x_5$

$N_{36} = x_7 + 0 \bullet 0 = x_7$

$N_{37} = x_6 + 0 \bullet x_1 = x_6$

$N_{41} = 0 + 0 \bullet 0 = 0$

$N_{43} = 0 + 0 \bullet 0 = 0$

$N_{44} = 1 + 0 \bullet 0 = 1$

$N_{45} = 0 + x_4 \bullet 0 = 0$

$N_{46} = x_8 + 0 \bullet 0 = x_8$

$N_{47} = 0 + 0 \bullet 0 = 0$

$N_{51} = 0 + 0 \bullet 0 = 0$

$N_{53} = 0 + 0 \bullet 0 = 0$

$N_{54} = 0 + 0 \bullet 0 = 0$

$N_{55} = 1 + x_4 \bullet 0 = 1$

$N_{56} = 0 + 0 \bullet 0 = 0$

$N_{57} = x_9 + 0 \bullet 0 = x_9$

$N_{61} = 0 + 0 \bullet 0 = 0$

$N_{63} = 0 + 0 \bullet 0 = 0$

$N_{64} = 0 + 0 \bullet 0 = 0$

$N_{65} = 0 + x_4 \bullet 0 = 0$

$N_{66} = 1 + 0 \bullet 0 = 1$

$N_{67} = x_{10} + 0 \bullet 0 = x_{10}$

$N_{71} = 0 + 0 \bullet 0 = 0$

$N_{73} = 0 + 0 \bullet 0 = 0$

$N_{74} = 0 + 0 \bullet 0 = 0$

$N_{75} = 0 + x_4 \bullet 0 = 0$

$N_{76} = 0 + 0 \bullet 0 = 0$

$N_{77} = 1 + 0 \bullet 0 = 1$

which gives the following reduced matrix

$$\begin{bmatrix} 1 & x_2 & x_3 & x_1x_4 & 0 & 0 \\ 0 & 1 & 0 & x_5 & x_7 & x_6 \\ 0 & 0 & 1 & 0 & x_8 & 0 \\ 0 & 0 & 0 & 1 & 0 & x_9 \\ 0 & 0 & 0 & 0 & 1 & x_{10} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

Now apply the equation (3.2) to the matrix (3.3) and we get. The removal of node 3. The new elements are

$N_{11} = 1 + x_2 \bullet 0 = 1$

$N_{14} = x_3 + 0 = x_3$

$N_{15} = x_1x_4 + x_2x_5 = x_1x_4 + x_2x_5$

$N_{16} = 0 + x_2x_7 = x_2x_7$

$N_{17} = 0 + x_2x_6 = x_2x_6$

$N_{41} = 0 + 0 \cdot 0 = 0$

$N_{44} = 1 + 0 \bullet 0 = 1$

$N_{45} = 0 + x_5 \bullet 0 = 0$

$N_{46} = x_8 + x_7 \bullet 0 = x_8$

$N_{47} = 0 + x_6 \bullet 0 = 0$

$N_{51} = 0 + 0 \bullet 0 = 0$

$N_{54} = 0 + 0 \bullet 0 = 0$

$N_{55} = 1 + x_5 \bullet 0 = 1$

$N_{56} = 0 + x_7 \bullet 0 = 0$

$N_{57} = x_9 + x_6 \bullet 0 = x_9$

$N_{61} = 0 + 0 \bullet 0 = 0$

$N_{64} = 0 + 0 \bullet 0 = 0$

$N_{65} = 0 + x_5 \bullet 0 = 0$

$N_{66} = 1 + x_7 \bullet 0 = 1$

$N_{67} = x_{10} + x_6 \bullet 0 = x_{10}$

$N_{71} = 0 + 0 \bullet 0 = 0$

$N_{74} = 0 + 0 \bullet 0 = 0$

$N_{75} = 0 + x_5 \bullet 0 = 0$

$N_{76} = 0 + x_7 \bullet 0 = 0$

$N_{77} = 1 + x_6 \bullet 0 = 1$

$$\begin{bmatrix} 1 & \mathrm{x}_3 & x_1x_4+x_2x_5 & x_2x_7 & x_2x_6 \\ 0 & 1 & 0 & x_8 & 0 \\ 0 & 0 & 1 & 0 & x_9 \\ 0 & 0 & 0 & 1 & x_{10} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

Now apply the equation (3.2) to the matrix (3.4) and we get the removal of node 4. The new elements are

$N_{11} = 1 + x_3 \bullet 0 = 1$

$N_{15} = x_1x_4 + x_2x_5 + x_30 = x_1x_4 + x_2x_5$

$N_{16} = x_2x_7 + x_3x_8 = x_2x_7 + x_3x_8$

$N_{17} = x_2x_6 + x_3 \bullet 0 = x_2x_6$

$N_{51} = 0 + 0 \bullet 0 = 0$

$N_{55} = 1 + 0 \bullet 0 = 1$

$N_{56} = 0 + x_8 \bullet 0 = 0$

$N_{57} = x_9 + 0 \bullet 0 = x_9$

$N_{61} = 0 + 0 \bullet 0 = 0$

$N_{65} = 0 + 0 \bullet 0 = 0$

$N_{66} = 1 + x_8 \bullet 0 = 1$

$N_{67} = x_{10} + 0 \bullet 0 = x_{10}$

$N_{71} = 0 + 0 \bullet 0 = 0$

$N_{75} = 0 + 0 \bullet 0 = 0$

$N_{76} = 0 + x_8 \bullet 0 = 0$

$N_{77} = 1 + 0 \bullet 0 = 1$

$$\begin{bmatrix} 1 & x_1x_4+x_2x_5 & x_2x_7+x_3x_8 & x_2x_6 \\ 0 & 1 & 0 & x_9 \\ 0 & 0 & 1 & x_{10} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

Now apply the equation (3.2) to the matrix (3.5) and we get the removal of node 5. The new elements are

$N_{11} = 1 + x_1x_4 + x_2x_5 \bullet 0 = 1$

$N_{16} = x_2x_7 + x_3x_8 + (x_1x_4 + x_2x_5) \bullet 0 = x_2x_7 + x_3x_8$

$N_{17} = x_2x_6 + (x_1x_4 + x_2x_5) \bullet x_9 = x_2x_6 + x_1x_4x_9 + x_2x_5x_9$

$N_{61} = 0 + 0 \bullet 0 = 0$

$N_{66} = 1 + 0 \bullet 0 = 1$

$N_{67} = x_{10} + x_9 \bullet 0 = x_{10}$

$N_{71} = 0 + 0 \bullet 0 = 0$

$N_{76} = 0 + 0 \bullet 0 = 0$

$N_{77} = 1 + x_9 \bullet 0 = 1$

$$\begin{bmatrix} 1 & x_2x_7 + x_3x_8 & x_2x_6 + x_1x_4\,x_9 + x_2x_5\,x_9 \\ 0 & 1 & x_{10} \\ 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

Now apply the equation (3.2) to the matrix (3.6) and we get the removal of node 6. The new elements are

$N_{11} = 1 + (x_2x_7 + x_3x_8) \bullet 0 = 1$

$N_{17} = x_2x_6 + x_1x_4x_9 + x_2x_5x_9 + (x_2x_7 + x_3x_8).\,x_{10}$

$\quad = x_2x_6 + x_1x_4x_9 + x_2x_5x_9 + x_2x_7\,x_{10} + x_3x_8\,x_{10}$

$N_{71} = 0 + 0 \bullet 0 = 0$

$N_{77} = 1 + x_{10} \bullet 0 = 1$

$$\begin{bmatrix} 1 & x_2x_6 + x_1x_4\,x_9 + x_2x_5\,x_9 + x_2x_7\,x_{10} + x_3x_8\,x_{10} \\ 0 & 1 \end{bmatrix} \tag{3.7}$$

From this final reduced matrix, the element N_{17} gives the transmission from node 1 (input) to node 7 (output) and, in this case is $x_2x_6 + x_1x_4\,x_9 + x_2x_5\,x_9\ + x_2x_7\,x_{10} + x_3x_8\,x_{10}$.

Boolean algebra was used in the reduction and evaluation of the reduced matrices, as well as in the transmission. The above transmission equation can be interpreted as follows using Boolean algebra rules: (x_2 and x_6) or (x_1 and x_4 and x_9) or (x_2 and x_5 and x_9) or (x_2 and x_7 and x_{10}) or (x_3 and x_8 and x_{10}). This expression reflects all feasible paths that exist between the system's input and output, and so are identical to the system's minimal paths or tie sets. From this point on, the system's reliability can be assessed using the assessment methodologies mentioned earlier in this chapter.

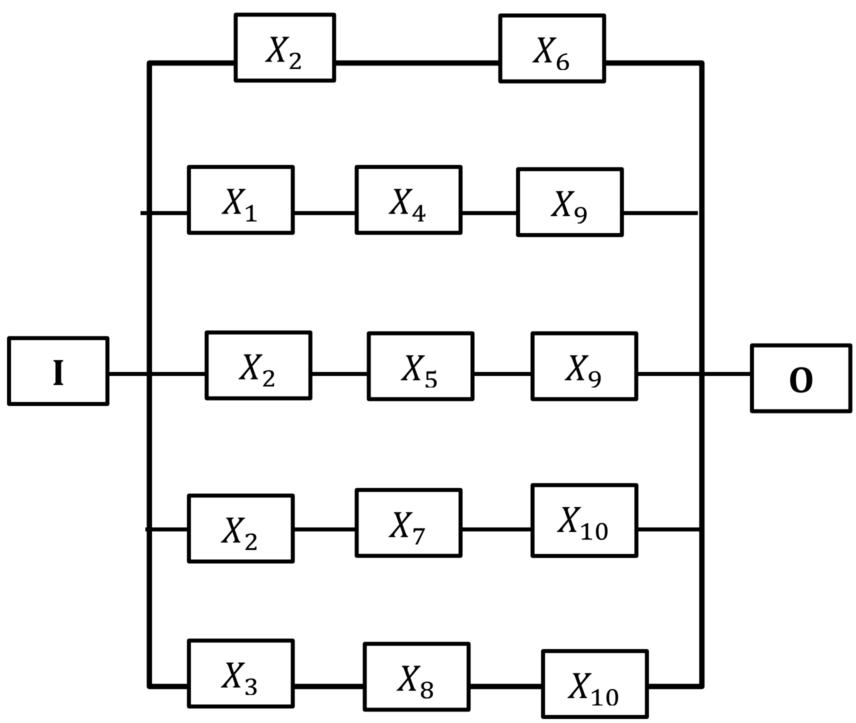


Figure 3.2: All minimal paths of bridge system.

3.2.2 Matrix Multiplication

The fundamental connection matrix is multiplied by itself a number of times till the resulting matrix is unchanged in this approach. In the present example, this multiplication process is as follows:

Let

$$M = \begin{bmatrix} node & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 1 & x_1 & x_2 & x_3 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & x_4 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 & x_5 & x_7 & x_6 \\ 4 & 0 & 0 & 0 & 1 & 0 & x_8 & 0 \\ 5 & 0 & 0 & 0 & 0 & 1 & 0 & x_9 \\ 6 & 0 & 0 & 0 & 0 & 0 & 1 & x_{10} \\ 7 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

then

$$M^2 = \begin{bmatrix} 1 & 2x_1 & 2x_2 & 2x_3 & x_1x_4 + x_2x_5 & x_2x_7 + x_3x_8 & x_2x_6 \\ 0 & 1 & 0 & 0 & 2x_4 & 0 & x_4x_9 \\ 0 & 0 & 1 & 0 & 2x_5 & 2x_7 & A \\ 0 & 0 & 0 & 1 & 0 & 2x_8 & x_8x_{10} \\ 0 & 0 & 0 & 0 & 1 & 0 & 2x_9 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2x_{10} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$A = 2x_6 + x_5x_9 + x_7x_{10}$

$$M^3 = \begin{bmatrix} 1 & 3x_1 & 3x_2 & 3x_3 & 3x_1x_4 + 3x_2x_5 & 3x_2x_7 + 3x_3x_8 & B \\ 0 & 1 & 0 & 0 & 3x_4 & 0 & 3x_4x_9 \\ 0 & 0 & 1 & 0 & 3x_5 & 3x_7 & C \\ 0 & 0 & 0 & 1 & 0 & 3x_8 & 3x_8x_{10} \\ 0 & 0 & 0 & 0 & 1 & 0 & 3x_9 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3x_{10} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where

B=$x_2x_6 + x_2(2x_6 + x_5x_9 + x_7x_{10}) + x_1x_4x_9 + x_3x_8x_{10}$.

C=$3x_6 + 3x_5x_9 + 3x_7x_{10}$

$$M^4 = \begin{bmatrix} 1 & 4x_1 & 4x_2 & 4x_3 & 6x_1x_4 + 6x_2x_5 & 6x_2x_7 + 6x_3x_8 & D \\ 0 & 1 & 0 & 0 & 4x_4 & 0 & 6x_4x_9 \\ 0 & 0 & 1 & 0 & 4x_5 & 4x_7 & E \\ 0 & 0 & 0 & 1 & 0 & 4x_8 & 6x_8x_{10} \\ 0 & 0 & 0 & 0 & 1 & 0 & 4x_9 \\ 0 & 0 & 0 & 0 & 0 & 1 & 4x_{10} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where

D=$2x_2x_6 + 2x_2(2x_6 + x_5x_9 + x_7x_{10}) + 2x_9(x_1x_4 + x_2x_5)$

$+2x_{10}(x_2x_7 + x_3x_8) + 2x_1x_4x_9 + 2x_3x_8x_{10}$.

E= $4x_6 + 6x_5x_9 + 6x_7x_{10}$

$$M^5 = \begin{bmatrix} 1 & 5x_1 & 5x_2 & 5x_3 & 10x_1x_4 + 10x_2x_5 & 10x_2x_7 + 10x_3X_8 & F \\ 0 & 1 & 0 & 0 & 5x_4 & 0 & 10x_4x_9 \\ 0 & 0 & 1 & 0 & 5x_5 & 5x_7 & G \\ 0 & 0 & 0 & 1 & 0 & 5x_8 & 10x_8x_{10} \\ 0 & 0 & 0 & 0 & 1 & 0 & 5x_9 \\ 0 & 0 & 0 & 0 & 0 & 1 & 5x_{10} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

F=$2x_2x_6 + 2x_2(2x_6 + x_5x_9 + x_7x_{10}) + 2x_2(4x_6 + 6x_5x_9$

$+6x_7x_{10}) + 2x_9(x_1x_4 + x_2x_5) + 2x_{10}(x_2x_7 + x_3x_8)$

$+8x_1x_4x_9 + 8x_3x_8x_{10}$.

G=$5x_6 + 10x_5x_9 + 10x_7x_{10}$

$$M^6=\begin{bmatrix} 1 & x_1 & x_2 & x_3 & x_1x_4+x_2x_5 & x_2x_7+x_3x_8 & H \\ 0 & 1 & 0 & 0 & x_4 & 0 & x_4x_9 \\ 0 & 0 & 1 & 0 & x_5 & x_7 & x_6+x_5x_9+x_7x_{10} \\ 0 & 0 & 0 & 1 & 0 & x_8 & x_8x_{10} \\ 0 & 0 & 0 & 0 & 1 & 0 & x_9 \\ 0 & 0 & 0 & 0 & 0 & 1 & x_{10} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$H=x_2x_6+x_1x_4x_9+x_2x_5x_9+x_2x_7x_{10}+x_3x_8x_{10}$.

Further powers of M have no effect on the resultant matrix, therefore the operation can be terminated here. The transmission from node 1 to node 7 is the same as in the node removal technique and the tie set method, as can be seen from the elements of M^6. When compared to the node removal approach, the advantage of the multiplication method is that it gives the transmission or tie sets between all pairs of nodes at the same time, whereas the node removal method only supplied the transmission between two nodes of interest. Despite the fact that the connection matrix approach It can also be considered a mechanism of deducing tie sets. Its a formal method in its own right. Several reliability evaluation approaches, as stated in Section 3.1, are fundamentally the same in concept; the only difference is how they are presented. In the case of tie sets and connection matrix approaches, this is particularly true.

3.3 Generation of Minimal Cut Sets

We will operate in this technique to using the set of all minimum paths, construct the incidence matrix of all minimal paths[62, 75]. Assume there are n minimum pathways, each of which is labeled by $p_1, p_2, , \ldots, p_n,$ The incidence matrix of all minimum pathways is then obtained. IM such that,

$$\mathrm{IM} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \quad (3.8)$$

where $a_{ij} \in \{0,1\}$ with $(i = 1,2,\ldots,m\,, j = 1,2,\ldots,n)$. A state is an n-dimensional array of 1_s and 0_s in which $a_{ij} = 1$ if and only if $x_i \in p_i$, else $a_{ij} = 0$.

The minimal cut sets must be generated in three steps. [17, 84].

1. If $\forall\, a_{ij} \neq 0$ of any column x_j of IM, then x_j forms a first order cut.

2. At the same time, combine two IM columns., If $\forall$i; $a_{ij} + a_{ik} \neq 0$ where $k > j$ (k = 1, 2, ..., n), then $x_j x_k$ Make a second cut in the same manner as the first. To give the second order minimal cuts, delete any cut that contains first order cuts. [13, 17, 20].

3. Repeat (2) with three columns at a moment to offer the third order reductions, this time remove any first and second order cuts and proceed until the highest cut order has been achieved.

Example 3.1 The bridge structure given by Fig**.(3.2)** .has the minimal path sets $p_1 = \{x_2 x_6\}, p_2 = \{x_1 x_4\, x_9\}$, $p_3 = \{x_2 x_5\, x_9\}$, $p_4 = \{x_2 x_7\, x_{10}\}, p_5 = \{x_3 x_8\, x_{10}\}$, Based on our descriptions above, the IM of the bridge system as follows,

$$\mathrm{IM} = \left[\begin{array}{c|cccccccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} \\ \hline P_1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ P_2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ P_3 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ P_4 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ P_5 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{array}\right] \quad (3.8)$$

Hence, no single column in IM exists in which none of the elements are zero, then there are no first order cuts. Then, applying Step (2) and also, there are no second order cuts. Then, applying Step (3) we found

$$a_{i1} + a_{i2} + a_{i3} \neq 0, a_{i1} + a_{i2} + a_{i8} \neq 0,$$
$$a_{i1} + a_{i2} + a_{i10} \neq 0, a_{i2} + a_{i3} + a_{i4} \neq 0,$$
$$a_{i2} + a_{i3} + a_{i9} \neq 0, a_{i2} + a_{i4} + a_{i8} \neq 0,$$
$$a_{i2} + a_{i4} + a_{i10} \neq 0, a_{i2} + a_{i8} + a_{i9} \neq 0,$$
$$a_{i2} + a_{i9} + a_{i10} \neq 0, a_{i6} + a_{i9} + a_{i10} \neq 0.$$

The results of the minimal cut sets are:

$\mathrm{Mcs}_1 = \{x_1, x_2, x_3\}, \mathrm{Mcs}_2 = \{x_1, x_2, x_8\},$

$\mathrm{Mcs}_3 = \{x_1, x_2, x_{10}\}, \mathrm{Mcs}_4 = \{x_2, x_3, x_4\},$

$\mathrm{Mcs}_5 = \{x_2, x_3, x_9\}, \mathrm{Mcs}_6 = \{x_2, x_4, x_8\},$

$\mathrm{Mcs}_7 = \{x_2, x_4, x_{10}\}, \mathrm{Mcs}_8 = \{x_2, x_8, x_9\},$

$\mathrm{Mcs}_9 = \{x_2, x_9, x_{10}\}, \mathrm{Mcs}_{10} = \{x_6, x_9, x_{10}\},$

$\mathrm{Mcs}_{11} = \{x_6, x_7, x_8, x_9\}, \mathrm{Mcs}_{12} = \{x_4, x_5, x_6, x_{10}\},$

$\mathrm{Mcs}_{13} = \{x_3, x_6, x_7, x_9\}, \mathrm{Mcs}_{14} = \{x_1, x_5, x_6, x_{10}\},$

$\mathrm{Mcs}_{15} = \{x_4, x_5, x_6, x_7, x_8\}, \mathrm{Mcs}_{16} = \{x_3, x_4, x_5, x_6, x_7\},$

$$\mathrm{Mcs}_{17} = \{x_1, x_5, x_6, x_7, x_8\}, \mathrm{Mcs}_{18} = \{x_1, x_3, x_5, x_6, x_7\}.$$

3.4 Path Tracing Method (PTM)

Every paths from a starting point to a finishing point is regarded with this technique[14, 21]. Since system success involves having at least one path from one end of the reliability block diagram to the other, as long as at least one path is available from the start to the end of the path, the system has not failed. Using this method to calculate the reliability of the complex system involves following the next steps [43, 75]

1. List all the minimum paths (tie-set) of the system.
2. Success of all parts in a minimal paths (tie-set) leads to system success.
3. This means interconnections in series between these components.
4. Each set of minimal tie generates the system accomplishment.
5. This implies that the minimum tie sets have parallel connections.
6. Draw equal system and use series and parallel reduction to calculate reliability of the system.

The system's reliability is the probability of uniting these minimum paths[24, 35]. General equation for union of minimal paths Pi can be given as

$$p_r\ (p_1\ +\ p_2 + \ldots + p_n) = p_r\left(\sum_{i=1}^{n} p_i\right)$$

$$= \sum_{i=1}^{n} p_r\ (p_i) - \sum_{i<j=2}^{n} p_r\left(p_i . p_j\right) +$$

$$\sum_{i<j<k=3}^{n} p_r\left(p_i . p_j p_k\right) - \cdots + \ (-1)^{n+1} p_r(p_1 . p_2 \ldots p_n)$$

This expression is also known as the inclusion-exclusion expansion [62]. For n independent minimal paths, the probability that at least one minimal path will be working. As a result we get

$$p_r\left(\sum_{i=1}^{n} p_i\right) = 1 - \left[[1\ -\ p_r(p_1)] \times [1 - p_r(p_2)] \times \ldots \times [1 - p_r(p_n)]\right] \quad (3.9)$$

Proposition 3.1.If $R_1; R_2; R_3; R_4; R_5; R_6; R_7; R_8; R_9; R_{10}$ are reliabilities of arcs (paths) in a complex system fig.(3.1), then the reliability Rs(t) of the system .

Rs = $R_2R_6 + R_1R_4R_9 + R_2R_5R_9 + R_2R_7R_{10} + R_3R_8R_{10} - R_2R_5R_6R_9 -$

$R_2R_6R_7R_{10} - R_1R_2R_4R_5R_9 - R_1R_2R_4R_6R_9 - R_2R_3R_6R_8R_{10} -$
$R_2R_3R_7R_8R_{10} - R_2R_5R_7R_9R_{10} + R_1R_2R_4R_5R_6R_9 -$
$R_1R_2R_4R_7R_9R_{10} - R_1R_3R_4R_8R_9R_{10} + R_2R_3R_6R_7R_8R_{10} -$
$R_2R_3R_5R_8R_9R_{10} + R_2R_5R_6R_7R_9R_{10} + R_1R_2R_4R_5R_7R_9R_{10} +$
$R_1R_2R_4R_6R_7R_9R_{10} + R_2R_3R_5R_6R_8R_9R_{10} + R_2R_3R_5R_7R_8R_9R_{10} +$
$R_1R_2R_3R_4R_5R_8R_9R_{10} + R_1R_2R_3R_4R_6R_8R_9R_{10} +$
$R_1R_2R_3R_4R_7R_8R_9R_{10} - R_1R_2R_4R_5R_6R_7R_9R_{10} -$
$R_2R_3R_5R_6R_7R_8R_9R_{10} - R_1R_2R_3R_4R_5R_6R_8R_9R_{10} -$
$R_1R_2R_3R_4R_5R_7R_8R_9R_{10} - R_1R_2R_3R_4R_6R_7R_8R_9R_{10} +$
$R_1R_2R_3R_4R_5R_6R_7R_8R_9R_{10}$. (3.10)

Proof. By using path tracing method, we have. Path sets : $p_1 = \{x_2x_6\}$, $p_2 = \{x_1x_4\ x_9\}$, $p_3 = \{x_2x_5\ x_9\}$, $p_4 = \{x_2x_7\ x_{10}\}$, $p_5 = \{x_3x_8\ x_{10}\}$, and then Fig. (3.1), which will indicate the parallel system's reliability [4], as follows: We'll make the assumption that $R_I(t)$ indicates the reliability of the p_I th component in a path set (the likelihood that the component Pi will be functional over the entire interval [0,t]). P_j , j $\in$ {1,2,3,4,5}. As a result, there are five possible path sets, each of which is represented as a parallel system in fig.(3.1), and each success that occurs in a path set causes the system to succeed. The BF (Boolean Function) Technique is used to analyze a symbolic expression for the reliability of such a complicated system. The probability that each path set P_j is:

$P_1 = R_2(t)R_6(t)$ (3.11)

$P_2 = R_1(t)\ R_4(t)\ R_9(t)$

$P_3 = R_2(t)\ R_5(t)\ R_9(t)$

$P_4 = R_2(t)R_7(t)\ R_{10}(t)$

$P_5 = R_3(t)R_8(t)\ R_{10}(t)$

The expression on the right-hand side of (3.11) is often written as $\coprod_{i=1}^{n} P_i$.

Then the reliability of the system is

$R_s = 1 - (1 - P_1)(1 - P_2)(1 - P_3)(1 - P_4)(1 - P_5) = \coprod_{i=1}^{5} P_i$, (3.12).

And by solving the above equation we will get the reliability polynomial (3.10). where the computations make sense in a probabilistic-boolean way, i.e., $R_i^n(t)$ is formally replaced by $R_i(t)$. We find the expression (3.10) which is of the reliability polynomial

3.5 Minimal Cut Method

We use the following assumptions to calculate the polynomial reliability by using this method [34, 38]

1. List all the system's minimal cut-sets.
2. In a minimal cut-set, failure of all parts creates system failure.
3. This means that these parts have parallel links.
4. Each set of minimal cutting creates failure of the system.
5. This means links between the minimal cut sets in sequence.
6. Use parallel and series decrease to calculate system and system reliability

$$R = 1 - \mathrm{p_r}(\mathrm{c_1} + \mathrm{c_2} + \ldots + \mathrm{c_n}). \qquad (3.13)$$

where $\mathrm{C}_i (i = 1, 2, \ldots, n)$ indicates the occurrence where all of the components in minimum cut set I fail [43], and n is the total number of minimal cut sets[48,75]. Equation (3.13) can be evaluated by applying the inclusion - exclusion rule, which is

$$pr(\mathrm{c_1} + \mathrm{c_2} + \cdots + \mathrm{c_n}) = \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{p_r}\,(\mathrm{c_i}) - \sum_{\mathrm{i<j=2}}^{\mathrm{n}} \mathrm{p_r}\,(\mathrm{c_i}.\mathrm{c_j}) + \sum_{\mathrm{i<j<k=3}}^{\mathrm{n}} \mathrm{p_r}\,(\mathrm{c_i}.\mathrm{c_j}\mathrm{c_k}) - \cdots + (-1)^{\mathrm{n+1}} \mathrm{p_r}(\mathrm{c_1}.\mathrm{c_2} \ldots \mathrm{c_n}).$$

For n independent minimal cuts, the probability that at least one minimal cut will be working [5]. As a result we get

$$p_r\left(\sum_{i=1}^{n} c_i\right) = 1 - \big[[1 - p_r(c_1)] \times [1 - p_r(c_2)] \times \ldots \times [1 - p_r(c_n)]\big] \quad (3.14)$$

Example 3.2 compute the system reliability depending on the sets of all minimal cut. From equation (3.14) and the sets of all minimal cut

$Mcs_1 = \{x_1, x_2, x_3\}, Mcs_2 = \{x_1, x_2, x_8\},$

$Mcs_3 = \{x_1, x_2, x_{10}\}, Mcs_4 = \{x_2, x_3, x_4\},$

$Mcs_5 = \{x_2, x_3, x_9\}, Mcs_6 = \{x_2, x_4, x_8\},$

$Mcs_7 = \{x_2, x_4, x_{10}\}, Mcs_8 = \{x_2, x_8, x_9\},$

$Mcs_9 = \{x_2, x_9, x_{10}\}, Mcs_{10} = \{x_6, x_9, x_{10}\},$

$Mcs_{11} = \{x_6, x_7, x_8, x_9\}, Mcs_{12} = \{x_4, x_5, x_6, x_{10}\},$

$Mcs_{13} = \{x_3, x_6, x_7, x_9\}, Mcs_{14} = \{x_1, x_5, x_6, x_{10}\},$

$Mcs_{15} = \{x_4, x_5, x_6, x_7, x_8\}, Mcs_{16} = \{x_3, x_4, x_5, x_6, x_7\},$

$Mcs_{17} = \{x_1, x_5, x_6, x_7, x_8\}, Mcs_{18} = \{x_1, x_3, x_5, x_6, x_7\}.$

$$\begin{aligned}
Rs = {} & [1 - (1 - x_1)(1 - x_2)(1 - x_3)] \times [1 - (1 - x_1)(1 - x_2)(1 - x_8)] \\
& \times [1 - (1 - x_1)(1 - x_2)(1 - x_{10})] \times [1 - (1 - x_2)(1 - x_3)(1 - x_4)] \\
& \times [1 - (1 - x_2)(1 - x_3)(1 - x_9)] \times [1 - (1 - x_2)(1 - x_4)(1 - x_8)] \\
& \times [1 - (1 - x_2)(1 - x_4)(1 - x_{10})] \times [1 - (1 - x_2)(1 - x_8)(1 - x_9)] \\
& \times [1 - (1 - x_2)(1 - x_9)(1 - x_{10})] \times [1 - (1 - x_6)(1 - x_9)(1 - x_{10})] \\
& \times [1 - (1 - x_6)(1 - x_7)(1 - x_8)(1 - x_9)] \\
& \times [1 - (1 - x_4)(1 - x_5)(1 - x_6)(1 - x_{10})] \\
& \times [1 - (1 - x_3)(1 - x_6)(1 - x_7)(1 - x_9)] \\
& \times [1 - (1 - x_1)(1 - x_5)(1 - x_6)(1 - x_{10})] \\
& \times [1 - (1 - x_4)(1 - x_5)(1 - x_6)(1 - x_7)(1 - x_8)] \\
& \times [1 - (1 - x_3)(1 - x_4)(1 - x_5)(1 - x_6)(1 - x_7)]
\end{aligned}$$

$$\times [1 - (1 - x_1)(1 - x5)(1 - x_6)(1 - x_7)(1 - x_8)]$$

$$\times [1 - (1 - x1)(1 - x3)(1 - x5)(1 - x_6)\ (1 - x_7)]. \qquad (3.15)$$

And by solving the above equation (3.15), we will get the reliability polynomial (3.10)

3.6 Reduction to Series Elements

This technique [43, 48] includes replacing each parallel path systematically with an equivalent single path and eventually lowering the scheme to one composed of only series components, as shown in Fig.3.1 .[54, 65].

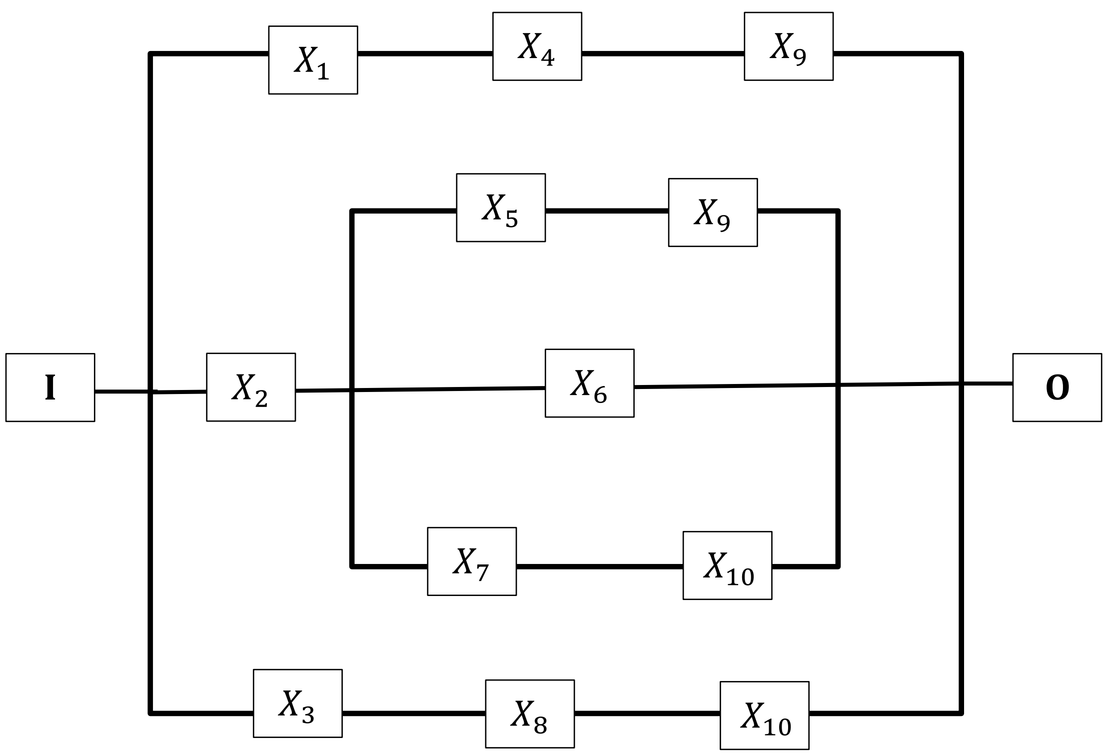


Figure 3.3: After reduction complex system.

Reduce [89,93], the given system to one consisting of only series element: Now, to solve the system

$$M_1 = 1 - [1 - R_1 R_4 R_9]$$

$$M_3 = 1 - [(1 - R_3 R_8 R_{10})]$$

$$N_1 = 1 - [(1 - R_2)]$$

$$N_2 = 1 - [(1 - R_5\, R_9\,)\,(1 - R_6)\,(1 - R_7 R_{10})]$$

$$M_2 = N_1 N_2$$

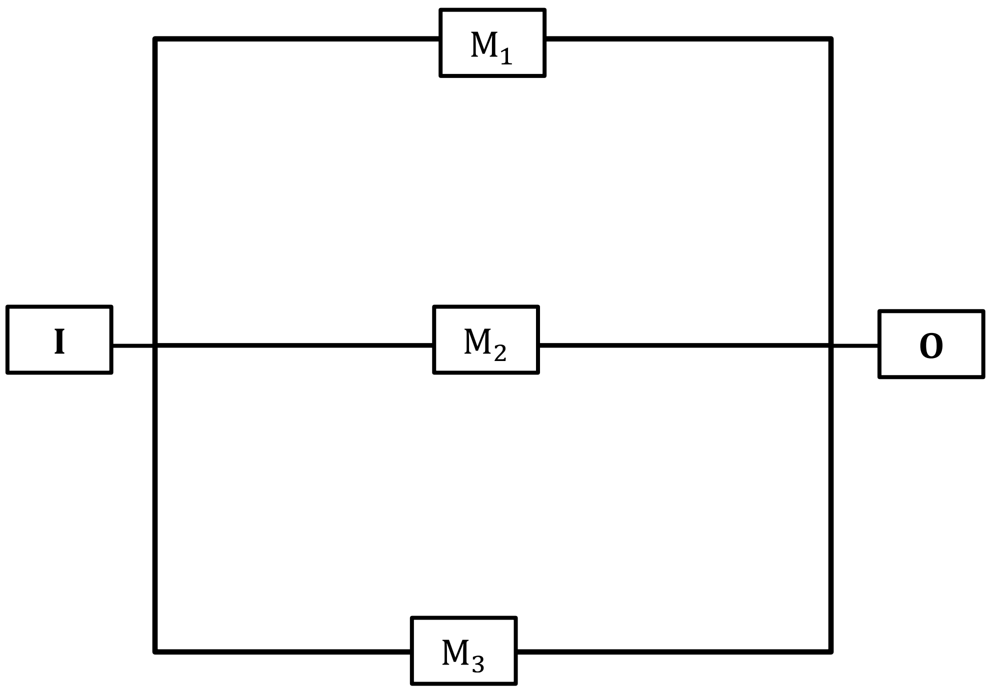


Figure 3.4: After reduction complex system.

$$\text{Rs} = 1 - [(1 - M_1)(1 - M_2)(1 - M_3)]$$

$$\text{Rs} = 1 - [(1 - R_1 R_4 R_9)(1 - R_3 R_8 R_{10})(1 - R_2(1 -$$

$$[(1 - R_5\, R_9)(1 - R_6)\,(1 - R_7 R_{10})]\,))] \qquad (3.16)$$

And by solving the above equation (3.15), we will get the reliability polynomial (3.10).

3.7 Inclusion- Exclusion Method

Inclusion–exclusion (IE) is a traditional approach for calculating a general system's reliability expression using its minimal routes. The IE approach, also known as Poincar'e or Sylvester's theorem, uses Bonferroni inequalities to produce successive upper and lower bounds on system reliability that converge to the exact system dependability[62].

Assume that E_j is the event that causes all components in the minimal path T_j to function. We can also state that E_j represents the event that occurs when the minimal path T_j is followed. The likelihood that the shortest path T_j will work can be stated as

$$P_r(E_j) = \prod_{i \in T_j} P_i \qquad (3.17)$$

If and only if at least one of the minimum paths works, a system with l minimal paths works. In other words, system success is linked to the occurrence. $\bigcup_{j=1}^{l} E_{j=1}$. The probability of the union l events, namely, the system's reliability, is equal to

$$\text{Rs} = \left(\bigcup_{j=1}^{l} E_{j=1} \right). \qquad (3.18)$$

and

$$S_k = \sum_{1 \le i_1 < \cdot\ \cdot\ \cdot < i_k \le l} \Pr(E_{i1} \cap E_{i2} \cap \ldots \cap E_{ik}). \qquad (3.19)$$

Then, S_k indicates the probability that any k minimum pathways are active at the same time. The system's reliability, which is equal to the probability of the union of the l minimum pathways, can be represented as follows using the IE principle (see Feller [75]).

$$\text{Rs} = \sum_{k=1}^{l} (-1)^{k-1} S_k. \qquad (3.20)$$

In application of equation (3.19), S_1 is included, S_2 is excluded, S_3 is included, S_4 is excluded, S_5 is included, and so forth. The name of the IE technique is derived from this. Upper and lower bounds on Rs become accessible as a result of this procedure of incorporating and omitting additional terms, as seen below:

$$Rs \leq S_1, \tag{3.21}$$

$$Rs \geq S_1 - S_2, \tag{3.22}$$

$$Rs \leq S_1 - S_2 + S_3, \tag{3.23}$$

$$Rs \geq S_1 - S_2 + S_3 - S_4, \tag{3.24}$$

$$Rs \leq S_1 - S_2 + S_3 - S_4 + S_5, \tag{3.25}$$

The so-called Bonferroni inequalities are these inequalities. These successive inequalities and, eventually, the actual value of Rs provide tighter restrictions on Rs, is obtained when $(-1)^{l-1}S_1$ is included. In reality, it may be sufficient to calculate only the first few S_k values in order to achieve an accurate Rs value. This is because S_1 has $\binom{n}{1}$ terms, S_2 has $\binom{n}{2}$ terms, . . . , S_n has $\binom{n}{2}$ terms. The IE approach appears to be ineffective for evaluating the reliability of a parallel system. The simple formula in equation can be used to assess the reliability of a parallel system (2.3). The maximum number of terms generated by the IE method for a system with l minimum routes is 2^l-1. Whenever some minimum pathways have common components, some of the 2^l-1 Because of the alternating signs in front of the possible terms of the IE approach, they cancel each other out. S_k for $1 \leq k \leq l$. As a result, the actual number of final terms generated by the IE approach is frequently substantially lower than the number generated by the other methods 2^l-1. However, the IE method has to evaluate all these 2^l-1 to get the final result, and then have some of the terms cancel each other. To put it another way, for systems with a large number of minimum pathways, this strategy is not very efficient.

Proposition 3.2. If $R_1; R_2; R_3; R_4; R_5; R_6; R_7; R_8; R_9; R_{10}$ are reliabilities of arcs (paths) in a complex system fig.(3.1), then the reliability $R_S(t)$ of the system (by using The Inclusion- Exclusion Method) is the reliability polynomial (3.10).

Proof. path sets: $p_1 = \{x_2 x_6\},\ p_2 = \{x_1 x_4\ x_9\},\ p_3 = \{x_2 x_5\ x_9\},$

$$p_4 = \{x_2 x_7\ x_{10}\},\ p_5 = \{x_3 x_8\ x_{10}\}.$$

For the graph in Fig. 1, let $p_i = p_r\{R_i = 1\}$ and $q_i = p_r\{R_i = 0\}$, minpaths using $MP_1 = \{x_2, x_6\}, MP_2 = \{x_1, x_4, x_9\}, MP_3 = \{x_2, x_5, x_9\},$

$$MP_4 = \{x_2, x_7, x_{10}\}, MP_5 = \{x_3, x_8, x_{10}\},$$

Then the system reliability Rs is:

$$S_1 = \{\Pr(MP_1) + \Pr(MP_2) + \Pr(MP_3) + \Pr(MP_4) + \Pr(MP_5)\}.$$
$$= x_2x_6 + x_1x_4x_9 + x_2x_5x_9 + x_2x_7x_{10} + x_3x_8x_{10}.$$

$$S_2 = \{\Pr(MP_1MP_2) + \Pr(MP_1MP_3) + \Pr(MP_1MP_4) + \Pr(MP_1MP_5)$$
$$+ \Pr(MP_2MP_3) + \Pr(MP_2MP_4(+\Pr(MP_2MP_5) + \Pr(MP_3MP_4))$$
$$+ \Pr(MP_3MP_5) + \Pr(MP_4MP_5)\}.$$

$$= x_1x_2x_4x_6x_9 + x_2x_5x_6x_9 + x_2x_6x_7x_{10} + x_2x_3x_6x_8x_{10}$$
$$+x_1x_2x_4x_5x_9 + x_1x_2x_4x_7x_9x_{10} + x_1x_3x_4x_8x_9x_{10} + x_2x_5x_7x_9x_{10}$$
$$+x_2x_3x_5x_8x_9x_{10} + x_2x_3x_7x_8x_{10}$$

$$S_3 = \{\Pr(MP_1MP_2MP_3) + \Pr(MP_1MP_2MP_4) + \Pr(MP_1MP_2MP_5)$$
$$+ \Pr(MP_1MP_3MP_4) + \Pr(MP_1MP_3MP_5) + \Pr(MP_1MP_4MP_5)$$
$$+ \Pr(MP_2MP_3MP_4) + \Pr(MP_2MP_3MP_5) + \Pr(MP_2\mathrm{MP4}MP_5)$$
$$+ \Pr(MP_3MP_4MP_5)\}.$$

$$= x_1\, x_2x_4x_5x_6x_9 + x_1x_2x_4x_6x_7x_9x_{10} + x_1x_2x_3x_4x_6x_8x_9x_{10}$$
$$+x_2x_5x_6x_7x_9x_{10} + x_2\, x_3x_5x_6x_8x_9x_{10} + x_2x_3x_6x_7x_8x_{10}$$
$$+x_1x_2x_4x_5x_7x_9x_{10} + x_1x_2x_3x_4x_5x_8x_9x_{10}$$
$$+x_1x_2x_3x_4x_7x_8x_9x_{10} + x_2x_3x_5x_7x_8x_9x_{10}$$

$$S_4 = \{\Pr(MP_1MP_2MP_3MP_4) + \Pr(MP_1MP_2MP_3MP_5)$$
$$+ \Pr(MP_1MP_2MP_4MP_5) + \Pr(MP_1MP_3MP_4MP_5)$$
$$+Pr(MP_2MP_3MP_4MP_5)\}.$$

$$= x_1x_2x_4x_5x_6x_7x_9x_{10} + x_1x_2x_3x_4x_5x_6x_8x_9x_{10}$$
$$+x_1x_2x_3x_4x_6x_7x_8x_9x_{10} + x_2x_3x_5x_6x_7x_8x_9x_{10}$$
$$+ x_1x_2x_3x_4x_5x_7x_8x_9x_{10}.$$

$$S_5 = \{\Pr(MP_1MP_2MP_3MP_4\, MP_5)\}.$$
$$= x_1\, x_2x_3x_4x_5x_6\, x_7x_8x_9x_{10}$$

$$\text{Rs} = S_1 - S_2 + S_3 - S_4 + S_5 \qquad (3.26)$$

And by solving the above equation (3.26), we will get the reliability

Chapter Four

Reliability Allocation and Importance

4.1 Introduction

Engineers are frequently faced with the problem of building a system that adheres to a set of dependability criteria while developing a new product[5]. The engineer is given the system's aim and must then create a design that will accomplish the system's required dependability while fulfilling all of the system's intended functions at the lowest possible cost. This entails a delicate balancing act of determining how to assign dependability to the system's components in order for the system to fulfill its reliability target while also meeting all of the other performance requirements. The construction of a model that reflects the complete system is the first step in the reliability optimization process[48]. This is achieved by creating a system reliability block diagram that depicts the reliability linkages between the system's components. The system dependability impact of various component adjustments may be calculated and assessed using this model, as well as the costs associated with implementing such alterations. The optimum combination of component reliability improvements that meet or exceed the performance requirements at the lowest cost may then be found using an optimization study for this problem[97]. In this chapter, we discuss the problem of allocating reliability to complex system after reducing the system in three components in parallel that became a subsystem to the system as shown in the fig.(3.4) in chapter three. In addition, the methods used to solve the problem of allocation have been modified from series system to parallel system, as in the above system. The purpose of reliability allocation is to establish a goal or objective for the reliability of each component so that the manufactures can have an idea of the performance required of third [104].

4.2 Reliability Allocation

Is the process by which the failure of a system is determined by using a logical method through the systems of the sub-system and its components. We define the system reliability objective of the individual components within the system that ensure access to the overall goal of system reliability[48]. For each for matting we use the component to refer to a typical unit or sub- system, which can be formulated in the allocation of reliability.

$$h(R_1^*, R_2^*, \ldots, R_n^*) \geq R^* \tag{4-1}$$

where R^* is the objective for system dependability, R_i^* i = (1, 2, ..., n) is the component i's reliability aim, and h represents a functional connection between system and component reliabilities[97]. The functional connection is derived from the preceding sections' system reliability analysis. Solving inequality eq.(4.1) for R_i^* is now the job of reliability allocation. In a complete reliability program, reliability allocation is a critical responsibility, especially when the goods under development are complex. The following are the different advantages of dependability allocation: A complex product includes a number of parts that are often planned, intended, tested and produced by numerous external providers and contractors as well as numerous inner departments[104]. It is essential that all parties concerned are partnered and share the goal of providing clients with the reliability needed to deliver the end product. In order to achieve this, each and every partner should be allocated and dedicated to the reliability goal from the point of perspective of project management. Reliability distribution for each element describes a legitimate objective of reliability.

1. Mandatory reliability demands force reliability duties to be regarded similarly in the process of product realization with engineering operations directed at meeting other client expectations, such as weight, price, and efficiency.
2. Allocation of reliability drives a profound knowledge of the hierarchical structure of the product (i.e. components, subsystems, and the final product's functional connections). The method may lead to weakness in design and subsequent enhancement being identified.
3. Reliability requirements are based on the mandatory reliability functions that must be taken into account on unequal to engineering activities designed to meet other customer ideas such as cost, performance and weight in the process of product realization.
4. Reliability assignment process outputs can be used as inputs for other reliability tasks. For example, a component assigned reliability is used to design a component reliability verification test.

4.3 Reliability Allocation Methods:

There are many method's to reliability allocation and these methods vary in complexity depending on how much the definition of the subsystem is available and the degree of accuracy required. In this part of the chapter we will look at the most common and used methods, such as

1. Equal Allocation Technique.
2. ARINC Approach method.
3. AGREE Allocation method.
4. Feasibility - of - objectives Technique method .

4.3.1 Equal Allocation Technique

The Equal Allocation Technique uses all standards for all system components and assigns a common reliability goal for all components to achieve the overall system reliability goal [48, 97,]. This method is simple and useful especially in the early design stage when no detailed information is available for the parallel system. Reliability of the system is the reliability results of the individual components[104].

Thus, can be written as:

$$1-\prod_{i=1}^{n}(1-R_i^*) \geq R^* \tag{4.2}$$

A component's minimal reliability requirement is determined by

$$R_i^* = (1-(1-R^*)^{\frac{1}{n}}), \quad \text{for} \quad i=1,2,\ldots,n \tag{4.3}$$

Where

R^* is the system's necessary dependability

R_i^* is subsystem i responsible for the reliability requirement

and each subsystem is equally reliable

if all components are exponential ,(4.2) then

$$\sum_{i=1}^{n} \lambda_i^* \leq \lambda^* \tag{4.4}$$

Where

λ^* is the maximum allowable failure rate to system

λ_i^* is the maximum allowable failure rate of component i and

$$\lambda_i^* = \frac{\lambda^*}{n}, \quad i= 1,2,3,\ldots,n. \tag{4.5}$$

Example 4.1 The parallel subsystem consists of three components as shown in Fig. (3.4) The lifetime of all subsystems of allocation is distributed evenly over components. If the above system reliability target at 48 months in service is 0.96, Determine the current reliability requirement as well as each subsystem's maximum permitted failure rate.

Solution

From eq.(4.3), the reliability of each individual subsystem is

$$R_i^*(48) = (1-(1-0.96))^{\frac{1}{3}} = 0.986\ ,\ i = 1,2,3.$$

The maximum allowable failure rate of the vehicle in accordance with the overall reliability target is

$$\lambda^* = -\frac{\ln[R^*(48)]}{48} = -\frac{\ln(0.96)}{48} = 8.5\times10^{-4} \text{ failures per month.}$$

From eq.(4.5), each individual subsystem's maximum allowed failure rate is

$$\lambda_i^* = \frac{\lambda^*}{n}\ ,\ i = 1,2,3$$

$$\lambda_i^* = \frac{8.5\times10^{-4}}{3} = 2.83\times10^{-4} \quad \text{failures per month, } i = 1,2,3.$$

4.3.2 ARINC Approach Method

The ARINC assumes that all parts are parallel linked, independent of each other, distributed exponentially, and have a common task time[48]. The job of determining individual failure rates becomes reliability allocation components λ_i^*such that eq.(4.4) is satisfied. The chance of component failure is included into the λ_i^* calculation. Using the following weighting variables:

$$w_i = \frac{\lambda_i}{\sum_{i=1}^{n}\lambda_i}\quad ,\ i = 1,2,\ldots,n \qquad (4.6)$$

where λ_i is the failure rate of component i as determined by historical data or forecast. The variables represent the probability of failure. The greater the w_i value, the more likely the component will fail[97]. As a result, the failure rate goal assigned to a component should be proportionate to the weight value:

$$\lambda_i^* = w_i \lambda_0, \quad i = 1, 2, ..., n. \tag{4.7}$$

where λ_0 is a constant. Because $\sum_{i=1}^{n} w_i = 1$ and if the equality holds in eq.(4.4), inserting eq.(4.7) into (4.4) yields $\lambda_0 = \lambda^*$. Therefore, eq.(4.7) can be written as

$$\lambda_i^* = w_i \lambda^*, \quad i = 1, 2, ..., n. \tag{4.8}$$

This is the component's maximum allowed failure rate[104]. The relevant dependability objective may be computed easily as follows:

$$R_i^* (t) = \exp(-w_i \lambda_i^* t), \qquad i = 1, 2, \ldots, n. \tag{4.9}$$

Example 4.2 See example (4.1). Warranty data complex subsystem. The failure rate estimates have generated, for the system above, the first component $\lambda_1 = 0.03$, the second component $\lambda_2 = 0.04$ and the third component $\lambda_3 = 0.05$ failures per month respectively. To reach a total reliability objective of 0.99, determine the reliability requirements at 32 months in operation and the maximum permitted failure rate for each subsystem.

Solution:

by equation (4.6) we get:

$$w_1 = \frac{\lambda_i}{\sum_{i=1}^{3} \lambda_i} = \frac{0.03}{(0.03)+(0.04)+(0.05)} = 0.25,$$

$$w_2 = \frac{0.04}{(0.03)+(0.04)+(0.05)} = 0.33,$$

$$w_3 = \frac{0.05}{(0.03)+(0.04)+(0.05)} = 0.41,$$

Now, we find $\lambda^*$where

$$\lambda^* = -\frac{\ln[R^*(32)]}{32} = -\frac{\ln(0.99)}{32} = 0.000314$$

By using equation (4.8) we get:

$$\lambda_1^* = 0.25 \times 0.000314 = 7.85 \times 10^{-5},$$

$$\lambda_2^* = 0.33 \times 0.000314 = 1.03 \times 10^{-4},$$

$$\lambda_3^* = 0.41 \times 0.000314 = 1.28 \times 10^{-4},$$

Now, we find the reliability per component by using equation (4.9) we get:

$R_1^*\ (32) = exp(-\lambda_1^* \times 32) = exp(-7.85 \times 10^{-5} \times 32) = 0.99,$

$R_1^*(32) = exp(-\lambda_1^* \times 32) = exp(-1.03 \times 10^{-4} \times 32) = 0.99,$

$R_3^*\ (32) = exp(-\lambda_1^* \times 32) = exp(-1.28 \times 10^{-4} \times 32) = 0.99,$

As a check, at 32 months, the resultant dependability is

$R_s^* = R_1^* + R_2^* + R_3^* - R_1^* R_2^* - R_1^* R_3^* - R_2^* R_3^* + R_1^* R_2^* R_3^*$

$= 0.99 + 0.99 + 0.99 - 0.99 \times 0.99 - 0.99 \times 0.99 - 0.99 \times 0.99$

$+ 0.99 \times 0.99 \times 0.99$

$= 0.99.$

4.3.3 AGREE Allocation Method

The AGREE allocation technique, established by the Electronic Equipment Reliability Advisory Group (AGREE), determines the minimum allowable mean failure time for each subsystem to meet the system reliability goal [48]. This strategy to distribution explicitly requires into account subsystem complexity. Complexity is described in terms of modules and their related circuitry, where an equal failure rate is assumed for each module. When defining the limit of modules, this hypothesis should be held in mind. In particular, module counts should be decreased for extremely secure subsystems like pcs because the failure rates are much smaller than those of less secure subsystems like actuators[97]. The AGREE allocation technique also takes into account the significance of individual subsystems, where significance is described as the likelihood of system failure if a subsystem fails. The significance represents the subsystem's essential to system achievement. The significance of 1 implies that for the system to operate the subsystem must work effectively. The significance of 0 implies that the subsystem's failure has no impact on the system's operation. Assume the subsystems are distributed, separately, exponentially, and function in sequence as to their impact on system achievement. You can then rewrite eq.(4.1) as

$$R^* (t) = \prod_{i=1}^{n} [1-w_i (1-R_i^* (t_i))] \tag{4.10}$$

where $R^* (t)$ is the system reliability target at time t_i, $R_i^* (t_i)$ the reliability target allocated for subsystem i at time t_i ($t_i \leq t$), w_i the importance of subsystem i, and n the number of subsystems[104]. It can be seen that the method of allocation allows a subsystem's mission time to be less than the system's. Since the times to failure of subsystems are distributed exponentially and we have the approximation eq.(4.9) can be written as

$$\sum_{i=1}^{n} \lambda_1^* w_i t_i = \ln[R^*(t)], \tag{4.11}$$

where λ_i^* is the failure rate allocated to subsystem i. Taking the complexity into account λ_i^* can be written as

$$\lambda^*_i = -\frac{m_i \ln(R^*(t))}{m\, w_i t_i} \quad , i= 1,2,3,\ldots\ldots,n \qquad (4.12)$$

Where

m_i is the number of modules in subsystem i.

w_i is the importance of subsystem i.

t_i is the operating time of subsystem i.

m is the total number of modules in the system $= \sum_{i=1}^{n} m_i$.

$$R_i^*(t_i) = 1 - \frac{1-[R^*(t)]^{\frac{m_i}{m}}}{w_i}, \qquad (4.13)$$

It can be seen that eq.(4.10) and eq.(4.11) for a small subsystem would result in a very low reliability goal. A very tiny w_i value distortedly outweighs the complexity impact and results in an unfair distribution. The technique only operates well when each subsystem's significance is close to 1.

Example 4.3 The above fig.(3.4) system consists of three sub-components. If the failure of the three components leads to a system failure, the reliability of the subsystems is required to achieve the system reliability target of 0.92 at a time of 10 hours.

Table (4-1) data for AGREE Reliability allocation

Number subsystem	Number of models mi	Importance w_i	Operating time t_i
1	9	1	10
2	12	0.9	10
3	6	1	6

Solution

total number of models m= 9+12+6= 27

By using equation (4.12) we find the failure rate to the subsystems (failures per hour) as

$$\lambda^*_i = -\frac{m_i \ln (R^*(t))}{m\, w_i\, t_i}$$

$$\lambda^*_1 = -\frac{9\times \ln(0.92)}{27\times1\times10} = 0.00278$$

$$\lambda^*_2 = -\frac{12\times \ln(0.92)}{27\times0.9\times10} = 0.00412$$

$$\lambda^*_3 = -\frac{6\times \ln(0.92)}{27\times1\times 6} = 0.00309$$

now, we find the reliability of sub system's using the equation (4.13)

$$R^*_i\,(t_i) = 1 - \frac{1-[R^*(t)]^{\frac{m_i}{m}}}{w_I},$$

$$R^*_i\,(10) = 1 - \frac{1-\ (0.92)^{\frac{9}{27}}}{1} = 0.972$$

$$R^*_i\,(10) = 1 - \frac{1-\ (0.92)^{\frac{12}{27}}}{1} = 0.959$$

$$R^*_i\,(6) = 1 - \frac{1-\ (0.92)^{\frac{6}{27}}}{1} = 0.982$$

We find the reliability system according to its equation (4.10):

$R^*(t) = \prod_{i=1}^{n}\ [\,1-w_i\ (1-R^*_i\ (t_i))]$ we get:

$R^*(10)$ = [1-1 (1-0.972)] [1-0.9 (1-0.959)] [1-1 (1-0.982)]

$R^* = 0.92$

goal reliability is required.

4.3.4 Feasibility of Objectives Technique

This approach was created to allocate mechanical and electrical system dependability without the need for repair[48]. Subsystem allocation factors, numerical complexity system classifications, performance time, state of the art, and environmental circumstances are determined using this approach[97]. The classification of this method is estimated on a scale from 1 to 10, with values determined as discussed:

1. The complexity of the system. The complexity is assessed by looking at the components that make up the system. The least complicated system is given a score of one, while the most complex system is given a score of ten. Performance time. The component that runs for the duration of task 10 is rated, the component that runs the least time during the task is at 1.
2. State-of-the-art. The lowest design or method is 10, and the most sophisticated is to assign a value of 1.
3. Environment. Components are also classified according to environmental conditions from 10 to 1. Components facing extremely harsh environments during their operation are expected to be classified as 10, which are expected to experience the least harsh environments classified as 1.

Each subsystem classification will be between 1 and 10 normalized so that a total is 1[104]. The basic equations in this method are:

$$\lambda_S = \sum \overline{\lambda_k} \tag{4.14}$$

$$\overline{\lambda_k} = \overline{C_k} . \lambda_S \tag{4.15}$$

Where λ_S is system failure rate.

T Is mission duration.

$\overline{C_k}$ Complexity of subsystem (k)

$\overline{\lambda_k}$ Is failure rate allocated to each subsystem.

as that

$$\overline{C_k} = \frac{w_K}{W} \quad (4.16)$$

$$\overline{W}_k = \bar{r}_{1k}.\bar{r}_{2k}.\bar{r}_{3k}.\bar{r}_{4k} \quad (4.17)$$

And

$$\overline{W} = \sum_{i=1}^{n} \overline{w}_k \quad (4.18)$$

Where

$\overline{W}$ is sum of the rated product's

$\overline{w}_k$ is rating for subsystem (k)

$\bar{r}_{ik}$ is rating for every of the four factors for each subsystem.

Example 4.4 A system of three sub-systems with a reliability of (0.91) at 96 hours. The engineering estimates of the complexity, state of the art, the time of performance and environment are given in table (4-2). Calculate the failure rate for each subsystem.

Table (4-2) values of subsystem allocation factors

Number of subsystem	Complexity	State of the art	Time of performance	Environment
1	5	6	5	5
2	7	6	10	2
3	6	5	5	5

Solution

by using equation (4.17) we get

$\overline{W}_k = \bar{r}_1.\bar{r}_2.\bar{r}_3.\bar{r}_4$

$\overline{W}_1$= (5) (6) (5) (5) = 750

$\overline{W}_2$= (7) (6) (10) (2) = 840

$\overline{W}_3$= (5) (6) (5) (5) = 750

Since

by equation (4.18) where

$\overline{W} = \sum_{i=1}^{3} \overline{w}_k$

So, $\overline{W}$ = 750+750+840

$\overline{W}$ = 2340

Now, we find C_k by using equation (4-16)

$\overline{C_1} = \frac{\overline{W}_1}{\overline{W}} = \frac{750}{2340} = 0.32$

$\overline{C_2} = \frac{\overline{W}_2}{\overline{W}} = \frac{840}{2340} = 0.359$

$\overline{C_3} = \frac{\overline{W}_3}{\overline{W}} = \frac{750}{2340} = 0.32$

$\lambda_S = \frac{-ln[R^*(t)\,]}{t} = \frac{-ln\,(0.91)}{96} = 0.00098$

Since

$\overline{\lambda_k} = \overline{C_k}.\,\lambda_S$

$\overline{\lambda_1}$= (0.32)(0.00098)= 0.000314

$\overline{\lambda_2}$= (0.359)(0.00098)= 0.000352

$\overline{\lambda_3}$= (0.32)(0.00098)= 0.000314

So, $\sum_{k=1}^{3} \overline{\lambda_k}$ = 0.00098.

This achieves equation (4.14) because of

$\lambda_S = \sum_{k=1}^{n} \overline{\lambda_k} = 0.00098$

And $\sum_{k=1}^{n} \overline{C_k} = 1.$

Then:

$$\sum_{k=1}^{3} \overline{C_k} = 0.999 \backsimeq 1.$$

4.4 Component Reliability Importance

Once the reliability of a system has been determined, engineers are often faced with the task of identifying the least reliable component(s) in the system in order to improve the design. mostly The least dependable component in a series system has the greatest impact on the overall system dependability. In this situation, if the system's dependability is to be enhanced, the efforts should be focused on increasing the component's reliability first. It is straightforward to detect the weak components in basic systems, such as a series system. In more complicated systems, however, this becomes a tough process. For complex systems, the analyst need a quantitative method that allows them to identify and quantify the relevance of each system component.

One technique of determining the relative significance of each component in a system in terms of the system's overall dependability is to use reliability importance metrics. Leemis [66] defines the dependability significance of a component in a system of components as follows:

$$I_R(i) = \frac{\partial \, Rs}{\partial R_i} \tag{4.20}$$

\where:

- Rs is the system reliability.
- R_i is the component reliability.

The value of the reliability importance provided by equation above is determined by the component's dependability as well as its location in the system. The dependability importance in terms of a value for each component may be calculated using the equation above.

Example 4.5 Find the general formulas of importance for each units in Fig. (3.1)

Solution:

The results obtained for each component of the system were by applying the eq.(4.20) to the reliability function of the system and the results were as follows:

$\frac{\partial R_s}{\partial R_1}$ = $R_4R_9 - R_2R_4R_5R_9 - R_2R_4R_6R_9 - R_2R_4R_5R_6R_9 - R_2R_4R_7R_9R_{10} - R_3R_4R_8R_9R_{10} + R_2R_4R_5R_7R_9R_{10} + R_2R_4R_6R_7R_9R_{10} + R_2R_3R_4R_5R_8R_9R_{10} + R_2R_3R_4R_6R_8R_9R_{10} + R_2R_3R_4R_7R_8R_9R_{10} - R_2R_4R_5R_6R_7R_9R_{10} - R_2R_3R_4R_5R_6R_8R_9R_{10} - R_2R_3R_4R_5R_7R_8R_9R_{10} - R_2R_3R_4R_6R_7R_8R_9R_{10} + R_2R_3R_4R_5R_6R_7R_8R_9R_{10}$

$\frac{\partial R_s}{\partial R_2}$ = $R_6 + R_5R_9 + R_7R_{10} - R_5R_6R_9 - R_6R_7R_{10} - R_1R_4R_5R_9 - R_1R_4R_6R_9 - R_3R_6R_8R_{10} - R_3R_7R_8R_{10} - R_5R_7R_9R_{10} + R_1R_4R_5R_6R_9 - R_1R_4R_7R_9R_{10} + R_3R_6R_7R_8R_{10} - R_3R_5R_8R_9R_{10} + R_5R_6R_7R_9R_{10} + R_1R_4R_5R_7R_9R_{10} + R_1R_4R_6R_7R_9R_{10} + R_3R_5R_6R_8R_9R_{10} + R_3R_5R_7R_8R_9R_{10} + R_1R_3R_4R_5R_8R_9R_{10} + R_1R_3R_4R_6R_8R_9R_{10} + R_1R_3R_4R_7R_8R_9R_{10} - R_1R_4R_5R_6R_7R_9R_{10} - R_3R_5R_6R_7R_8R_9R_{10} - R_1R_3R_4R_5R_6R_8R_9R_{10} - R_1R_3R_4R_5R_7R_8R_9R_{10} - R_1R_3R_4R_6R_7R_8R_9R_{10} + R_1R_3R_4R_5R_6R_7R_8R_9R_{10}$

$\frac{\partial R_s}{\partial R_3}$ = $R_8R_{10} - R_2R_6R_8R_{10} - R_2R_7R_8R_{10} - R_1R_4R_8R_9R_{10} + R_2R_6R_7R_8R_{10} - R_2R_5R_8R_9R_{10} + R_2R_5R_6R_8R_9R_{10} + R_2R_5R_7R_8R_9R_{10} + R_1R_2R_4R_5R_8R_9R_{10} + R_1R_2R_4R_6R_8R_9R_{10} + R_1R_2R_4R_7R_8R_9R_{10} - R_2R_5R_6R_7R_8R_9R_{10} - R_1R_2R_4R_5R_6R_8R_9R_{10} - R_1R_2R_4R_5R_7R_8R_9R_{10} - R_1R_2R_4R_6R_7R_8R_9R_{10} + R_1R_2R_4R_5R_6R_7R_8R_9R_{10}$.

$\frac{\partial R_s}{\partial R_4}$ = $R_1R_9 - R_1R_2R_5R_9 - R_1R_2R_6R_9 + R_1R_2R_5R_6R_9 - R_1R_2R_7R_9R_{10} - R_1R_3R_8R_9R_{10} + R_1R_2R_5R_7R_9R_{10} + R_1R_2R_6R_7R_9R_{10} + R_1R_2R_3R_5R_8R_9R_{10} + R_1R_2R_3R_6R_8R_9R_{10} + R_1R_2R_3R_7R_8R_9R_{10} - R_1R_2R_5R_6R_7R_9R_{10} -$

$R_1R_2R_3R_5R_6R_8R_9R_{10}–R_1R_2R_3R_5R_7R_8R_9R_{10}–R_1R_2R_3R_6R_7R_8R_9R_{10} +$
$R_1R_2R_3R_5R_6R_7R_8R_9R_{10}$

$$\frac{\partial R_s}{\partial R_5} = R_2R_9– R_2R_6R_9–R_1R_2R_4R_9– R_2R_7R_9R_{10}+R_1R_2R_4R_6R_9– R_2R_3R_8R_9R_{10}+R_2R_6R_7R_9R_{10}+R_1R_2R_4R_7R_9R_{10}+R_2R_3R_6R_8R_9R_{10}+ R_2R_3R_7R_8R_9R_{10}+R_1R_2R_3R_4R_8R_9R_{10}– R_1R_2R_4R_6R_7R_9R_{10} – R_2R_3R_6R_7R_8R_9R_{10}–R_1R_2R_3R_4R_6R_8R_9R_{10}–R_1R_2R_3R_4R_7R_8R_9R_{10} +R_1R_2R_3R_4R_6R_7R_8R_9R_{10}$$

$$\frac{\partial R_s}{\partial R_6} = R_2–R_2R_5R_9–R_2R_7R_{10}–R_1R_2R_4R_9–R_2R_3R_8R_{10}+R_1R_2R_4R_5R_9+ R_2R_3R_7R_8R_{10}+R_2R_5R_7R_9R_{10}+R_1R_2R_4R_7R_9R_{10}+ R_2R_3R_5R_8R_9R_{10}+R_1R_2R_3R_4R_8R_9R_{10}–R_1R_2R_4R_5R_7R_9R_{10} – R_2R_3R_5R_7R_8R_9R_{10}–R_1R_2R_3R_4R_5R_8R_9R_{10}– R_1R_2R_3R_4R_7R_8R_9R_{10}+R_1R_2R_3R_4R_5R_7R_8R_9R_{10}$$

$$\frac{\partial R_s}{\partial R_7} = R_2R_{10}–R_2R_6R_{10}–R_2R_3R_8R_{10}–R_2R_5R_9R_{10}–R_1R_2R_4R_9R_{10}+ R_2R_3R_6R_8R_{10}+R_2R_5R_6R_9R_{10}+R_1R_2R_4R_5R_9 R_{10} + R_1R_2R_4R_6R_9R_{10}+R_2R_3R_5R_8R_9R_{10}+R_1R_2R_3R_4R_8R_9R_{10}– R_1R_2R_4R_5R_6R_9R_{10}–R_2R_3R_5R_6R_8R_9R_{10}– R_1R_2R_3R_4R_5R_8R_9R_{10}– R_1R_2R_3R_4R_6R_8R_9R_{10} + R_1R_2R_3R_4R_5R_6R_8R_9R_{10}$$

$$\frac{\partial R_s}{\partial R_8} = R_3R_{10}–R_2R_3R_6R_{10}–R_2R_3R_7R_{10}–R_1R_3R_4R_9R_{10} + R_2R_3R_6R_7R_{10} – R_2R_3R_5R_9R_{10}+R_2R_3R_5R_6R_9R_{10}+R_2R_3R_5R_7R_9R_{10} + R_1R_2R_3R_4R_5R_9R_{10} + R_1R_2R_3R_4R_6R_9R_{10} + R_1R_2R_3R_4R_7R_9R_{10}–R_2R_3R_5R_6R_7R_9R_{10}– R_1R_2R_3R_4R_5R_6R_9R_{10} – R_1R_2R_3R_4R_5R_7R_9R_{10} – R_1R_2R_3R_4R_6R_7R_9R_{10} + R_1R_2R_3R_4R_5R_6R_7R_9R_{10}.$$

$$\frac{\partial R_s}{\partial R_9} = R_1R_4+R_2R_5–R_2R_5R_6–R_1R_2R_4R_5–R_1R_2R_4R_6–R_2R_5R_7R_{10}+$$

$R_1R_2R_4R_5R_6 – R_1R_2R_4R_7R_{10} – R_1R_3R_4R_8R_{10} –$

$R_2R_3R_5R_8 R_{10} + R_2R_5R_6R_7 R_{10} + R_1R_2R_4R_5R_7 R_{10} +$

$R_1R_2R_4R_6R_7 R_{10} + R_2R_3R_5R_6R_8 R_{10} + R_2R_3R_5R_7R_8R_{10} +$

$R_1R_2R_3R_4R_5R_8 R_{10} + R_1R_2R_3R_4R_6R_8 R_{10} +$

$R_1R_2R_3R_4R_7R_8 R_{10} – R_1R_2R_4R_5R_6R_7 R_{10} –$

$R_2R_3R_5R_6R_7R_8 R_{10} – R_1R_2R_3R_4R_5R_6R_8 R_{10} –$

$R_1R_2R_3R_4R_5R_7R_8 R_{10} – R_1R_2R_3R_4R_6R_7R_8 R_{10} +$

$R_1R_2R_3R_4R_5R_6R_7R_8 R_{10}.$

$\frac{\partial R_s}{\partial R_{10}} = R_2R_7 + R_3R_8 – R_2R_6R_7 – R_2R_3R_6R_8 – R_2R_3R_7R_8 – R_2R_5R_7R_9 –$

$R_1R_2R_4R_7R_9 – R_1R_3R_4R_8R_9 + R_2R_3R_6R_7R_8 – R_2R_3R_5R_8R_9 +$

$R_2R_5R_6R_7R_9 + R_1R_2R_4R_5R_7R_9 + R_1R_2R_4R_6R_7R_9 +$

$R_2R_3R_5R_6R_8R_9 + R_2R_3R_5R_7R_8R_9 + R_1R_2R_3R_4R_5R_8R_9 +$

$R_1R_2R_3R_4R_6R_8R_9 + R_1R_2R_3R_4R_7R_8R_9 – R_1R_2R_4R_5R_6R_7R_9 –$

$R_2R_3R_5R_6R_7R_8R_9 – R_1R_2R_3R_4R_5R_6R_8R_9 –$

$R_1R_2R_3R_4R_5R_7R_8R_9 – R_1R_2R_3R_4R_6R_7R_8R_9 +$

$R_1R_2R_3R_4R_5R_6R_7R_8R_9$

Chapter Five
Optimization Algorithms

Optimization Algorithms

5.1 Introduction

Various swarm optimization algorithms have been presented since the early 60's, These algorithms contain Genetic (GA), ant colony (ACO), particle swarm optimization (PSO), and bee colony (BCO). These algorithms have established their potential to resolve various optimization problems [30].

5.2 Optimization Models

Decision making always have to take by scientists and engineers[30]. As the world grow increasingly competitive and complex, Decision making consists in the following steps as fig 5.1:

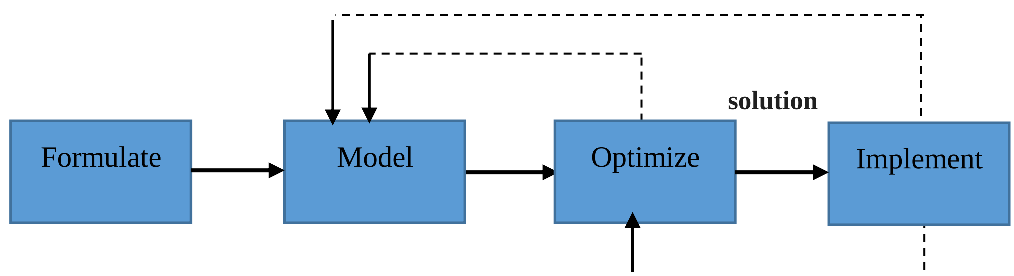


Fig.5.1 The decision making steps.

• **Formulate the problem:** identified the decision problem. Then, made early statement of the problem.

• **Model the problem:** built an abstract mathematical model for the problem.

• **Optimize the problem:** Once the problem is modeled, the solving procedure generates a "good" solution for the problem. The solution may be optimal or suboptimal.

• **Implement a solution:** The solution will be tested in practical way by the maker of decision and if it is “acceptable.” Will implement. If it is unacceptable, the optimization algorithm and/or the model has to be enhanced and repeated the process of decision making.

5.3 Intelligence Algorithms

These algorithms include Particle Swarm, Ant Colony Optimization, Genetic Algorithms and Artificial Bee Colony.

5.3.1 Genetic Algorithm

To maximize the solution of the issues, the ideas of biological processing of normal selection and survival of the fittest are used. Genetic processes, which is a population-based met heuristics, is a well-known approach. [30,31] J. Holland (University of Michigan) created this method in the 1970s to better understand natural system adaptation processes [23]. Then, in the 1980s,[52] they were used to machine learning and optimization.

Multiple pairs of solutions known as parents are altered in GAs to produce additional solutions known as children, who then have children as parents. The following stages and algorithm5.1 describe the general steps of GAs:

1. Make a pool of solutions at random.

2. select competitive parents from these solutions pool

3. use a crossover technique to create offspring from a pair of parents

4. Mutate those children in order to produce mutant youngsters.

5. choose a subset of the mutant children to replace the previous generation's parents.

6. continue until the termination condition is met.

Simple genetic algorithm procedure

Generate Initial Population

Evaluation Initial Population

repeat

Select Parent Solutions

Cross parent solution with probability

Mutate parent solution with probability

Insert child solutions in the new population

until Termination criterion : population is full

Algorithm 5.1 Simple genetic algorithm procedure

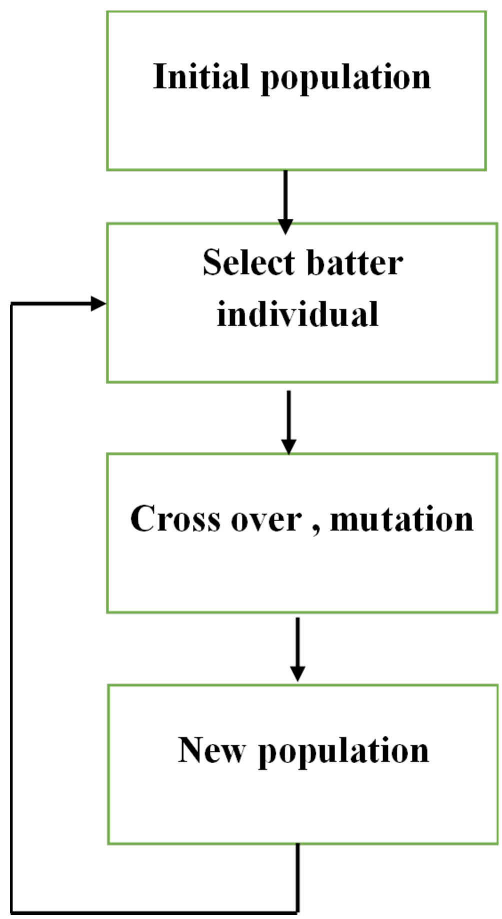


Fig 5.2 Genetic Algorithm Flowchart.

A set of chromosomes is referred to as a population. Each chromosome represents a potential solution for optimizing the problem's solutions. It is made up of a series of symbols known as "genes," which are used to find the best solution by using stochastic operators. To generate a generation, a met heuristic algorithm requires many phases: a selection phase, a replication phase, an assessment phase, and a replacement phase. During a series of generations, the population evolves until a stopping criteria is met. The following is a summary of the algorithm's key steps:

1. Coding

The process of representing individual genes is called Encoding. encoding can be achieved using numbers, bits, arrays, trees, lists or any other structures. The encoding based chiefly on solving the problem. For example as bellow, one can encode directly binary numbers. Chromosome1 11010001101, chromosome2 01111111100 [87]

2. Population

A population is a group of individuals. A population contains number of individuals being tested, some information about search space and the individuals defined by the phenotype parameters. For example the population in the following fig 5.3.

Population	Chromosome 1	1 1 1 0 0 0 1 0
	Chromosome 2	0 1 1 1 1 0 1 1
	Chromosome 3	1 0 1 0 1 0 1 0
	Chromosome 4	1 1 0 0 1 1 0 0

Fig 5.3 Population example .

3. Evaluation

The fitness function evaluates the parameters of each chromosome during the evaluation step. It's worth noting that each solution has a certain number of replications[86].

4. Selection

The process of selecting two parents from the population for crossing is called Selection. After an encoding, the following phase is to choose how to achieve selection. The upper the fitness function, the more chance an individual has to be selection.

5. Crossover

The process of taking two parent and creating from them a child is called Crossover. After the selection, the population is improved with enhanced individuals. To explain the crossover operator, the fallowing example in fig 5.4:

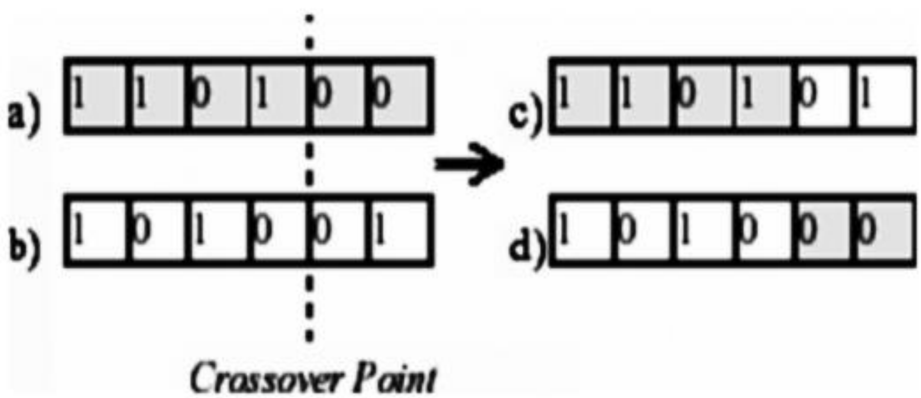


Fig 5.4 Example of Crossover.

6. Mutation

Mutation is a stochastic process that alters a person in order to generate a different one. The term "mutation" refers to the investigation of various regions of solution space. As a result, the population is more diverse, and the population does not decline to a local minimum. It's worth noting that this stage creates a new person from an existing one. For example, exhibit 5.5 [87] shows how to make a mutation by altering the assignment of task 3.

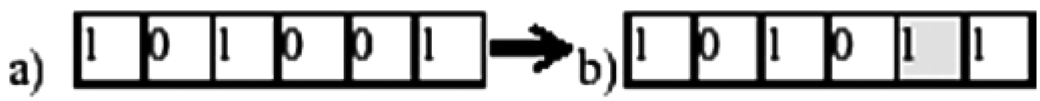


Fig 5.5 Mutation Example.

7. Stopping Criterion

Total computing time, number of iterations in which any improvement occurs sequentially, total number of iterations, or number of iterations without any improvement can all be used as termination criteria.

5.3.2 Particle Swarm Optimization

PSO algorithm is a technique of stochastic optimization based on swarm, which was suggested by Kennedy and Eberhart (1995). PSO algorithm simulates behavior of animal's social, with herds, insects, fishes and birds. Which is conform a cooperative way to find food, and each member in the

swarms saves varying the search pattern according to the learning experiences of its own and other member[29,85]. PSO To select a spot with adequate food, it replicates the social behavior of real creatures such as bird flocking and fish schooling. Indeed, without any central supervision, a coordinated behavior based on local motions arises in such swarms[74].

PSO was originally created to solve problems involving continuous optimization. In [49], the first application to optimization issues was presented.

5.3.2.1 Elements Used in PSO

First, we have start with the fundamentals concepts used in the PSO. we shall summary as brief concepts of the PSO elements:

Fitness Function: The Fitness function is the method for determining the best solution. It is usually an objective function..

Pbest : It's the particle's best location out of all the ones it's been to thus far.

Gbest : The position in which all of the particles visited so far have the best fitness.

Velocity Update: Velocity is a vector that determines the particle's speed and direction. The equation updates the velocity (1).

Position Update: Every particle tries to get into the ideal place for maximum fitness. To discover the global optimum, each particle in PSO changes its location. Equation (2) [48] is used to update position.

5.3.2.2 The Basic Idea of PSO

1 Avoid collisions with neighboring.

2 Stay near neighboring.

3 Match the velocity of neighboring.

The PSO algorithm shown in algorithm 5.2 and flowchart in fig 5.6

Algorithm 5.2 Template of the particle swarm optimization algorithm.
Random initialization of the whole swarm ;
Repeat
Evaluate f (xi) ;
For all particles i
Update velocities: $v_i(t) = v_i(t - 1) + \rho_1 \times (pbest_i - x_i(t - 1)) + \rho_2 \times (gbest - x_i(t - 1))$;…(5.1)
Move to the new position: $x_i(t) = x_i(t - 1) + v_i(t)$;…..(5.2)
If $f(x_i) < f(pbest_i)$ Then $pbest_i = x_i$;
If $f(x_i) < f(gbest)$ Then $gbest = x_i$;
Update(x_i, v_i) ;
End For
Until Stopping criteria

each particle must retain its pbest known as local best position and Gbest known as global best position. The particle's velocity and location are updated using equations (5.1) and (5.2).Where,

$V_i(t)$ **:** the new velocity.

$V_i(t-1)$ **:** the old velocity.

ρ_1 , ρ_2 **:** learining factors.

$pbest_i$ **:** The particle's best location among all the sites it has visited thus far.

gbest **:** The place where, among all the particles visited, the highest fitness is obtained, as in [30,31].

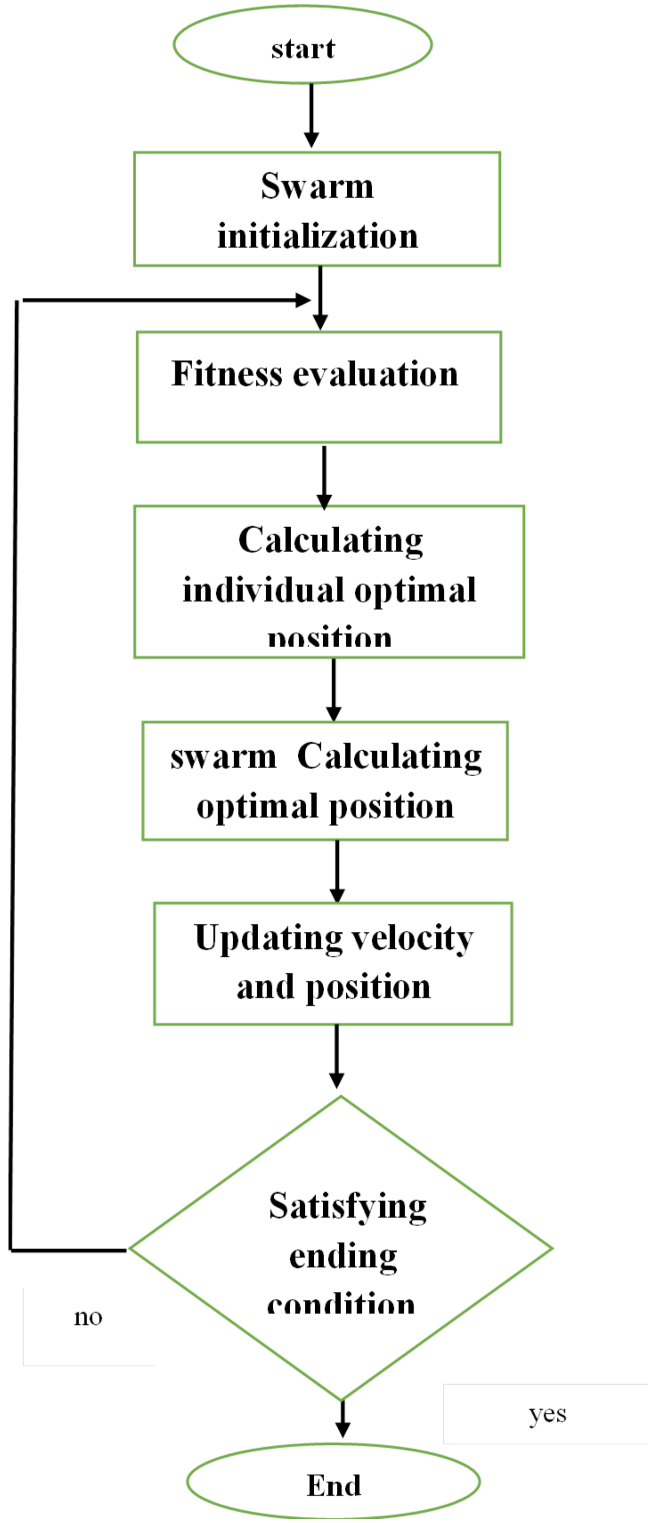


Fig 5.6 PSO algorithm flowchart.

They reduced each person to a particle with no mass or volume. with only velocity and position, so they called this algorithm “particle swarm optimization algorithm." On this basis, PSO algorithm can may be summed up as follows: The PSO algorithm is a swarm-based search method in which each person is referred to as a particle. Defined as a potential solution of the optimized problem in D-dimensional search space, and It can remember both the swarm's and its own optimum positions, as well as the velocity. In each generation, the particles information is combined to modify the velocity of each dimension, which is then used to calculate the particle's new location. Particles alteration their states constantly in the multi-dimensional search space, until they reach balance or optimal state, or beyond the calculating limits. Unique connection between dissimilar dimensions of the problem space is presented via the objective functions. Lots of empirical evidences have displayed that this algorithm is an actual optimization tool [30].

5.3.3 Ant Colony Optimization

The goal of ACO is to solve optimization issues by simulating the cooperative behavior of real ants. They may be thought of as multiagent systems, with each agent being inspired by the behavior of a real ant [30]. The basic principle behind genuine ant behavior is that ants use cooperative behavior to accomplish tasks like finding the shortest pathways to food sources and transporting the food. The shortest path between two places can be found by an ant colony. Figure 5.3 depicts Goss [85] .'s experiment with a genuine colony of Argentine ants. It's worth noting that ants have poor vision. The colony has access to a food supply that is connected to the colony's nest by two routes. During their journeys, they leave a chemical trail (pheromone) on the ground. The pheromone is a volatile olfactory molecule. The purpose of this path is to guide the other ants to the destination. The higher the concentration of pheromone on a path, the more likely the ants will choose it. The route for a given ant is determined by the amount of pheromone smelt. Furthermore, this chemical substance has a decreasing action over time (evaporation process), and the quantity left by one ant is based on the amount of food available (reinforcement process).

Algorithm 5.3 Template of the ACO
Initialize the pheromone trails ;
Repeat
For each ant **Do**
Solution construction using the pheromone trail ;
Update the pheromone trails:
Evaporation ;
Reinforcement ;
Until Stopping criteria
Output: Best solution found or a set of solutions.

As Fig. 5.7, When confronted with a barrier, each ant has an equal chance of choosing the left or right path. The ant will leave a higher amount of pheromone on the left path since it is shorter than the right one and so requires less travel time.

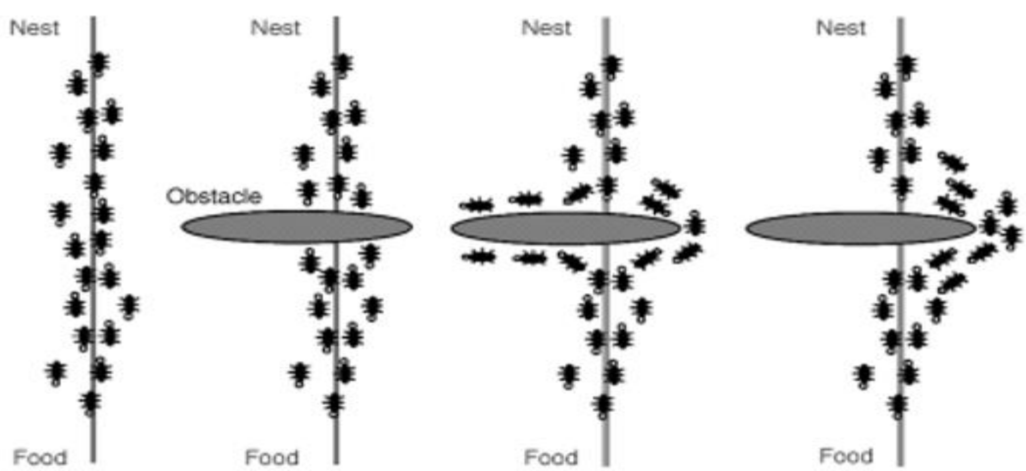


Fig. 5.7 facing an obstacle.

The greater the pheromone trail, the more ants follow the left path. As a result, the shortest path becomes apparent. The evaporation stage will emphasize this point. Stigmergy is the term for this type of indirect collaboration. Algorithm

5.3 is the ACO algorithm. The pheromone information is initialized at the start. The process is then broken down into two iterative steps: solution creation and pheromone updating.

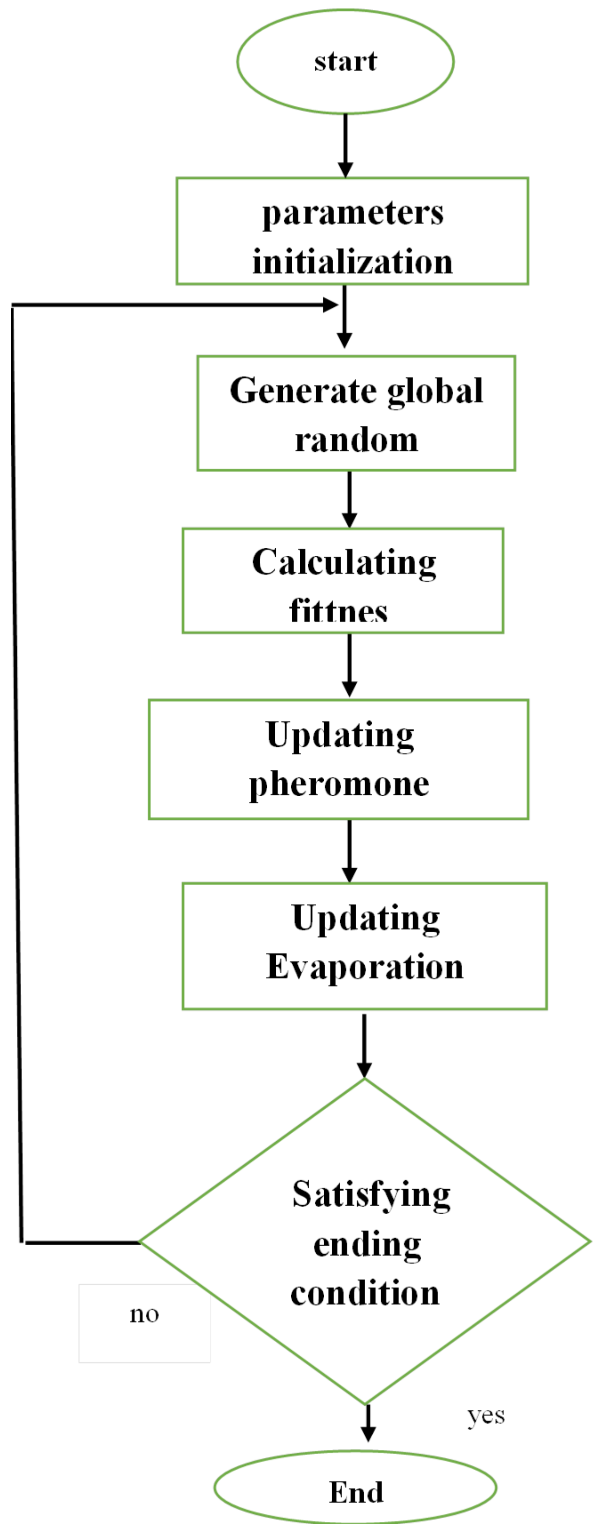


Fig 5.8 . flowchart of the ACO.

5.3.4 Bee Colony Optimization.

The Bee Colony Optimization (BCO) met heuristic has been introduced fairly recently by Lučić and Teodorović as a new direction in the field of Swarm Intelligence. The BCO has been successfully applied to various engineering and management problems by Teodorović and coauthors. Swarm intelligence methods include the bee colony optimization-based algorithm, which is a stochastic P-met heuristic. Many research based on different bee colony behaviors have been produced during the last decade to address complicated combinatorial or continuous optimization issues [86]. Bee colony optimization methods are based on the behavior of a honeybee colony, which has a number of characteristics that may be utilized as models for intelligent and collective action. Nectar exploration, mating in flight, food gathering, waggle dance, and division of work are among these characteristics. Three distinct theories underpin bee colony-based optimization algorithms: In the bee colony, food foraging, nest site seeking, and marriage are all common activities. Each model specifies a distinct task-related behavior.

5.3.4 .1 Bees in Nature

A bee colony typically consists of one reproductive female known as the queen, a few thousand males known as drones, and tens of thousands of sterile females known as workers. The queen produces numerous baby bees called broods after mating with multiple drones. The structural and functional distinctions between these four honeybee components are as follows:

Queen: A bee colony has a single queen who is the breeding female and has a life expectancy of 3 to 5 years.

Drones: Drones represent the males in the hive, which number between 300 and 3000. Drones are formed when the queen produces unfertilized eggs, and they fertilize with a receptive queen in the summer and, on rare occasions, in the fall. The drone has a 90-day life expectancy. After a successful mating, it dies.

Workers: Workers are female bees who do not have the ability to reproduce. During the cold season, they survive for 4 to 9 months and can number up to 30,000. However, they only survive for around 6 weeks during the summer, when their population swells to 80,000. The worker is in charge of defending the hive with its stinging stinger. As a result, after stinging, it dies.

Broods: Broods are the name given to the juvenile bees. They are hatched when the queen lays eggs in specific honeycomb chambers known as brood frames. The employees then apply royal jelly to the brood heads.

There are two techniques for finding a food source (see Table 5.1 for a comparison between natural and artificial bee colonies):

• **Exploration of food sources:** A scout bee searches the surrounding region for a food source. In the affirmative situation, it returns to the hive's dancing floor and performs a waggle dance to alert the nest members. Onlookers turn into recruit bees, which ultimately turn into hired foragers.

• **Exploitation of a food source:** A forager in the field assesses the number of food sources available and makes a selection based on nectar quality. Either it keeps exploitation going by memorizing the finest meal it's discovered so far, or it gives up. In such instance, utilizing probabilistic division, the employed bee becomes an unemployed scout or spectator bee.

TABLE 5.1 Analogy Between Natural and Artificial Bee Colonies.

Natural Bee Colony	Artificial Bee Colony
Food source	Solution
Quality of nectar	Objective function
Onlookers	Exploitation of search
Scout	Exploration of search

Algorithm 5.4 displays the bee algorithm's template, which was inspired by a bee colony's food seeking activity [30]. The method begins by randomly mapping n scout bees in the search space. The quality of the scout bees' visited places (i.e., solutions) is then assessed. After that, the fittest bees are picked, and the places they visit for neighborhood searches are chosen. The algorithm then runs searches in the vicinity of the selected locations, allocating additional observer bees to conduct searches in the vicinity of the best e sites. The bees can be chosen based on their fitness and the fitness of the places they are visiting. Alternatively. The fitness values are used to calculate the likelihood of the bees being chosen. Searches in the vicinity of the top e sites that offer more potential solutions are made more specific by enlisting the help of more bees than the other bees. This differential recruiting, along with scouting, is a crucial activity of the bee algorithm (BA). Only the most fit bees from each patch will be picked to produce the next bee population. There are no such limitations in nature. This limitation is in place to limit the number of solutions that may be investigated. The remaining bees in the population are then randomly assigned to search the

search space for additional potential solutions. This cycle is repeated until a predetermined stopping condition is satisfied. The colony will have two groups for its new population at the conclusion of each iteration: representatives from each selected patch and scout bees assigned to undertake random searches[30,86].

Algorithm 5.4 Template of the bee algorithm.
Random initialization of the whole colony of bees ;
Evaluate the fitness of the population of bees ;
Repeat /∗ Forming new population ∗/
Select sites for neighborhood search ;
Determine the patch size ;
Recruit bees for selected sites and evaluated their fitness ;
Select the representative bee from each patch ;
Assign remaining bees to search randomly and evaluate their
fitness ;
Until Stopping criteria

Algorithm 5.4 Template of the bees algorithm.

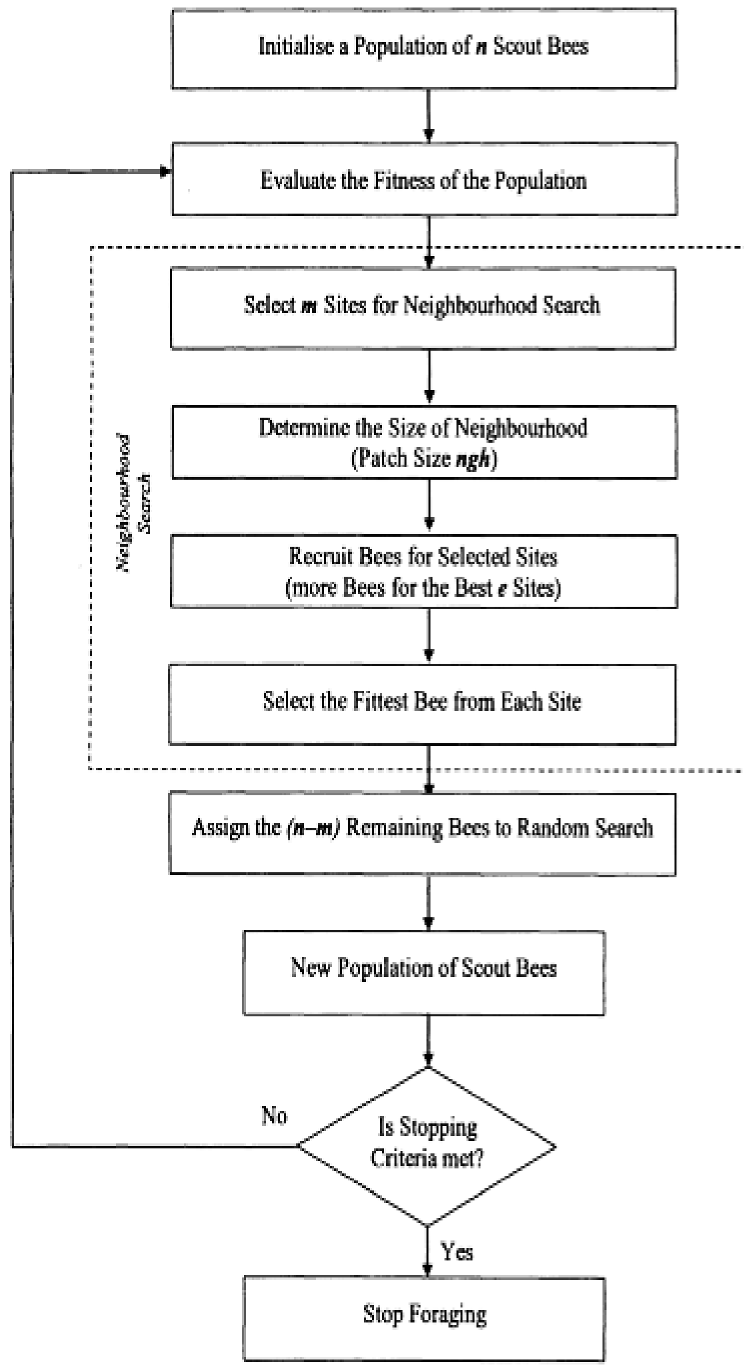


Fig 5.9 . flow chart of the BCO.

Chapter Six

A Comparative Study Between GA, PSO,ACO and BCO to Evaluate the Reliability Optimization Problem for Complex System

6.1 Introduction

In this chapter, the reliability allocation and optimization for each component of the was calculated[5,46]. Use algorithms (genetic algorithm ,particle swarm optimization, ant colony algorithm and bees colony optimization) to solve the allocation problem and optimization the reliability of the system, as well as calculate the total cost of the system[60,63]. The three cost functions are also used addressed (exponential behavior with feasibility factor model, exponential behavior model and logarithmic model). After solving the allocation problem, the reliability importance of each system component was calculated. The goal of this chapter was to compare between the results of (GA, PSO, ACO and BCO) in terms of reliability allocation and optimization, the total cost, reliability exact[64,70], reliability importance and then whichever more effective than another[77,98,106]

6.2 Optimization of Complex System

Consider a complex system composed of connected elements [6, 7, 70]. We use the statements: $0 \leq R_i \leq 1$ is the component reliability i ; $C_i(R_i)$ the component costs i ; $C(R_1, \ldots, R_n) = \sum_{i=1}^{n} a_i\, c_i$ (R_i) the system's total cost, where $a_i > 0$; R_s is the system reliability; R_G is system Reliability goal. Every part of the system has a unique functionality and there are many possibilities, many system parts give us the same functionality with different levels of reliability[1,2]. The aim is to achieve the allocation of reliability to some or all parts of the system. The Q question is included as a major problem in nonlinear programming [40,53], a cost-and-function that can be evaluated and is a nonlinear limit[3,4].

Q: Find

$$\text{Minimize } C(R_i, \ldots, R_n) = \sum_{i=1}^{n} a_i C_i(R_i),\ a_i > 0, \quad (6.1)$$

subject to

$$R_s \geq R_G,$$

$$0 \leq R_i < 1, i = 1, \dots, n$$

Assuming the partial cost function is reasonable $C_i(R_i)$ meets certain conditions[16] Increasing the positive, differentiable,

$\left[\Rightarrow \frac{dC_i}{dR_i} \geq 0\right].$

The foregoing formulation is designed to achieve a minimum total system cost, subject to R_G, a lower limit on the system reliability.

6.3 Cost Function

Changing a design always comes at a cost, whether it's due to a change in vendors, the usage of higher-quality materials, retooling expenditures, or administrative fees. Before attempting to enhance reliability, the cost as a function of reliability for each component must be calculated[77,98]. Otherwise, the design alterations could result in an overly expensive or overdesigned system. The engineer will be able to determine which components to improve and how to effectively concentrate effort and resources by developing the "cost of reliability" connection. The first step will be to establish a link between improved costs and reliability. Formulating the cost function from actual cost data is the preferable method. This can be done based on previous experience. The expenses associated with each stage of improvement can also be assessed if a reliability growth program is in place. Defining the various costs associated with various suppliers or component models is also helpful in developing a component cost as a function of dependability model. However, in many instances, no such information is accessible. As a result, to undertake reliability optimization, a generic (default) behavior model of cost vs component reliability was constructed. This function's goal is to simulate overall cost behavior for all types of components. Of course, it is impossible to create a model that is perfectly relevant in every circumstance; however, the proposed link is broad enough to encompass the majority of scenarios. In addition to the default model formulation, user specified cost models can be defined.

6.4 Application to Complex System

The complex system represented in fig.(3.1) has the same primary reliability for all its computers at specified times of 90% . The system reliability objective at a specified time is 90%[1,2]. The reliability polynomial of the given system was calculated by using the approach to minimal path [3,4,14], we will get the reliability polynomial (3.10).

6.5 Models of Three Cost Functions for Compute the Optimal Reliability Allocation Problem

6.5.1 Exponential Behavior Model with Feasibility Factor

Let $0 < f_i < 1$ be a feasibility factor [1,2], $R_{i,min}$ be minimum reliability and $R_{i,max}$ be maximum reliability[3,4]. Exponential behavior is another important cost function.

$$C_i(R_i) = \exp[(1-f_i)\frac{R_i - R_{i,min}}{R_{i,max} - R_i}],\ R_{i,min} \le R_i \le R_{i,max}\ , i = 1,2,\dots,n. \quad (6.2)$$

The optimization problem becomes:

$$Minimize\ C(R_i,\dots,R_n) = \sum_{i=1}^{n} a_i\ \exp[(1-f_i)\frac{R_i - R_{i,min}}{R_{i,max} - R_i}], i = 1,2,\dots,n.$$

Subject to:

$$\text{Rs} \ge R_G$$

$$R_{i,min} \le R_i < R_{i,max}\ , i = 1,\dots,n\ .$$

Table 6.1: Summary table for optimal reliability allocation by using GA, PSO, ACO and BCO with the exponential behavior model with feasibility factor

Components	Particle Swarm	Genetic Algorithm	Ant colony	Bees Colony
R_1	0.7975	0.9485	0.794	0.809
R_2	0.8437	0.9535	0.844	0.858
R_3	0.7812	0.9477	0.782	0.822
R_4	0.5145	0.9485	0.815	0.809
R_5	0.5452	0.9372	0.644	0.766
R_6	0.7923	0.9456	0.801	0.822
R_7	0.6523	0.9372	0.694	0.744
R_8	0.7162	0.9487	0.801	0.822
R_9	0.7960	0.9504	0.815	0.822
R_{10}	0.6884	0.9498	0.815	0.822
Rs	0.9019	0.9989	0.95	0.9616

The findings were obtained using the exponential behavior model with feasibility factor in the four algorithms (genetic algorithm, particle swarm optimization, ant colony algorithm and bees colony optimization) to improve and customize authenticity. It was found that by using this function, the best result was when using the genetic algorithm, where the reliability reached (0.9989) and followed by then bees colony optimization. The documented value was (0.9616) and then ant colony algorithm where the notarized value (0.95) finally particle swarm optimization where the swarm value was (0.9019).

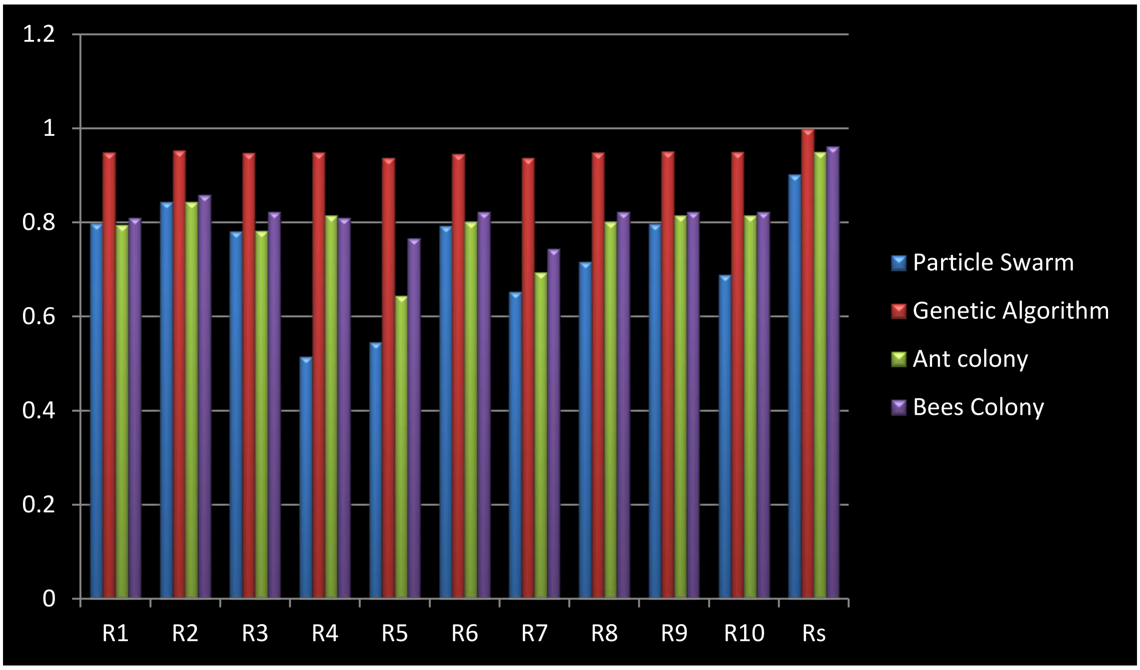


Figure 6.1: Reliability allocation by using GA, PSO, ACO and BCO with the exponential behavior model with feasibility factor.

6.5.2 The Exponential Behavior Model

Let $0 \leq R_i < 1, i = 1, \ldots, n$ and a_i, b_i , are constants, i=1,2,…,n. The most significant cost-function is exponential behavior[1,2]. It was suggested by the [3,4], in the form

$$C_i\ (R_i) = a_i e^{\left(\frac{b_i}{1-R_i}\right)}, a_i > 0\ , b_i > 0\ , \quad i = 1, \ldots, n \qquad (6.3)$$

The problem with the optimization becomes:

$$Minimize\ \ C(R_i, \ldots, R_n) = \sum_{i=1}^{n} a_i\ e^{\left(\frac{b_i}{1-R_i}\right)}, i = 1,2, \ldots, n.$$

Subject to :

$$\text{Rs}\ \geq R_G$$

$$0 \leq R_i < 1,\ \ i = 1, \ldots, n$$

Table 6.2: Summary table for optimal reliability allocation by GA, PSO, ACO and BCO with The exponential behavior model

Components	Particle Swarm	Genetic Algorithm	Ant colony	Bees Colony
R_1	0.7994	0.7914	0.786	0.795
R_2	0.8556	0.9900	0.843	0.853
R_3	0.6808	0.7856	0.794	0.779
R_4	0.5191	0.7893	0.811	0.795
R_5	0.6021	0.6836	0.664	0.674
R_6	0.6530	0.8111	0.792	0.776
R_7	0.7393	0.6807	0.689	0.713
R_8	0.7558	0.7851	0.777	0.795
R_9	0.8119	0.8382	0.811	0.795
R_{10}	0.7741	0.8320	0.811	0.795
Rs	0.9012	0.9845	0.947	0.946

The results were acquired by the use of to improve and modify authenticity, we employed a function the exponential behavior model in four algorithms (genetic algorithm, particle swarm optimization, ant colony algorithm, and bees colony optimization). It was discovered that while applying this function, the genetic algorithm produced the best results, with a dependability of (0.9845), followed by the ant colony method. The documented value was (0.947), and the notarized value for bee colony optimization was (0.946). Finally, particle swarm optimization was performed, with the swarm value being (0.9012).

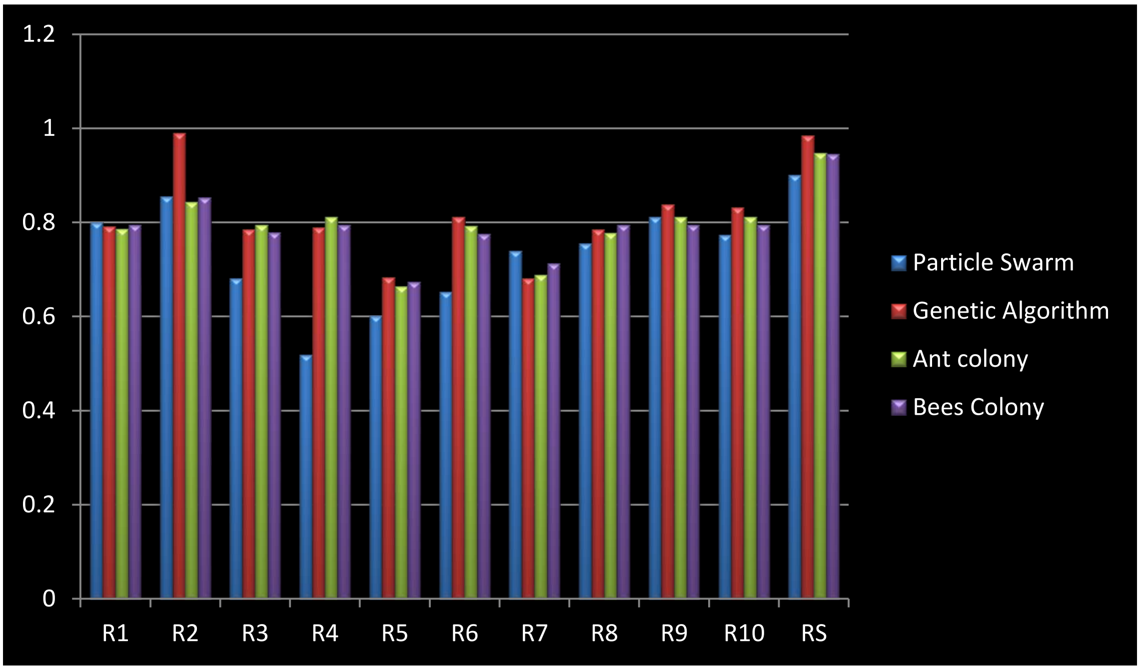


Figure 6.2 Reliability allocation by using GA, PSO, ACO and BCO with the given exponential behavior model.

6.5.3 The Logarithmic Model

Let $0 < R_i < 1, i = 1, \dots, n$ and a_i , are constants, i=,1,2,…,n. .It was proposed by [1,2,3,4], in the form

$$C_i(R_i) = a_i \ln\left(\frac{1}{1-R_i}\right), a_i > 0\ ,\ i = 1, \dots, n \qquad (6.4)$$

The optimization problem becomes:

$$Minimize\ \ C(R_i, \dots, R_n) = \sum_{i=1}^{n} a_i\ \ln\left(\frac{1}{1-R_i}\right),\ \ i = 1,2, \dots, n.$$

Subject to:

$$\text{Rs} \geq R_G$$

$$0 \leq R_i < 1, i = 1, \dots, n$$

Table 6.3: Summary table for optimal reliability allocation by using GA, PSO, ACO and BCO with the logarithmic model

Components	Particle Swarm	Genetic Algorithm	Ant colony	Bees Colony
R_1	0.671	0.5001	0.579	0.732
R_2	0.8961	0.9329	0.998	0.999
R_3	0.7368	0.5001	0.627	0.604
R_4	0.5138	0.5000	0.746	0.522
R_5	0.7413	0.5001	0.896	0.954
R_6	0.6321	0.8911	0.941	0.954
R_7	0.6858	0.5000	0.896	0.826
R_8	0.6742	0.5000	0.553	0.55
R_9	0.7666	0.5000	0.941	0.954
R_{10}	0.7845	0.5001	0.941	0.954
Rs	0.9001	0.9006	0.998	0.999

The results were acquired by the use of to increase and customize authenticity, we employed a logarithmic model function in four algorithms (genetic algorithm, particle swarm optimization, ant colony algorithm, and bees colony optimization). The best result was obtained when utilizing the bee colony optimization function, with a reliability of (0.999), and was followed by the ant colony algorithm. The documented value was (0.998), followed by a genetic algorithm with a notarized value of (0.9006). Finally, particle swarm optimization was performed, with the reliability value being (0.9001).

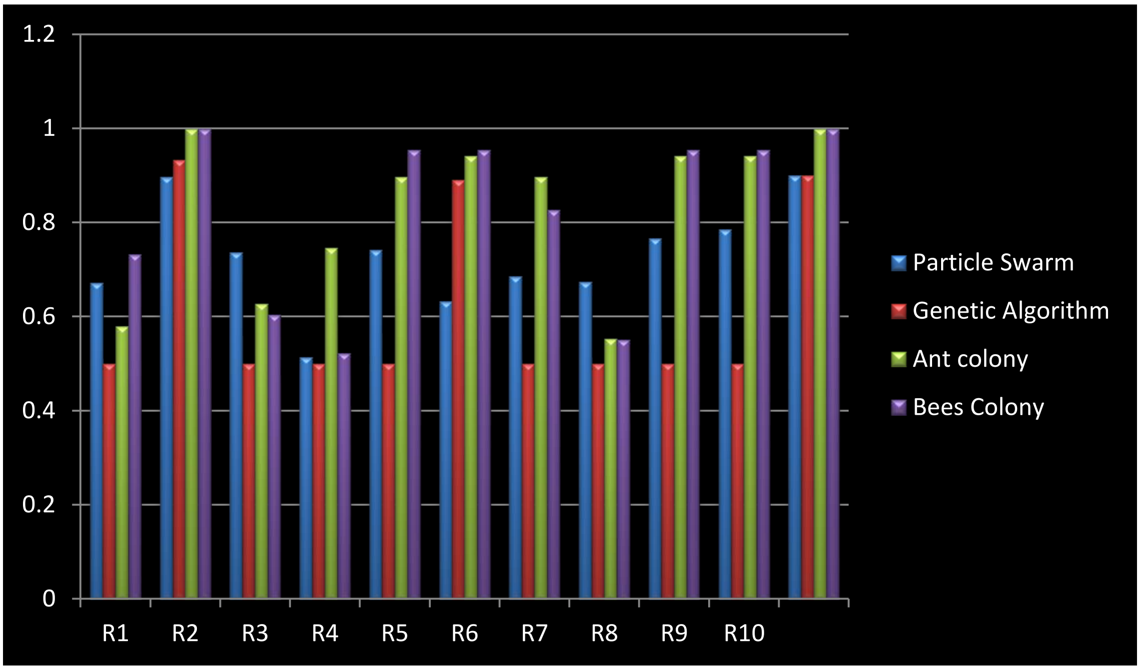


Figure 6.3: Reliability allocation by using GA, PSO, ACO and BCO with the given Logarithmic model.

6.6 Costing and Customization Using Genetic Algorithm

We used genetic algorithm and three cost functions (exponential behavior with feasibility factor model, exponential behavior model and logarithmic model) to obtain the highest reliability of the system with the lowest possible cost and to compare the results between the three cost functions by using GA in terms of reliability allocation and optimization, accurate reliability[4]. As the following table.

Table 6.4: Summary table for a comparative results exponential behavior with feasibility factor model, exponential behavior model and logarithmic model functions by using GA.

Components	Feasibility factor model	Cost.	Exponential behavior model	Cost.	Logarithmic model	Cost.
R_1	0.9485	654.8049	0.7914	7.4775	0.5001	0.0693
R_2	0.9535	919.8793	0.9900	50.0541	0.9329	0.2702
R_3	0.9477	572.6958	0.7856	6.6537	0.5001	0.0693
R_4	0.9485	654.8049	0.7893	7.1627	0.5000	0.0693
R_5	0.9372	143.7653	0.6836	1.7193	0.5001	0.0693
R_6	0.9456	412.2466	0.8111	11.7264	0.8911	0.2217
R_7	0.9372	143.7653	0.6807	1.6754	0.5000	0.0693
R_8	0.9487	677.6589	0.7851	6.5891	0.5000	0.0693
R_9	0.9504	919.879	0.8382	26.0453	0.5000	0.0693
R_{10}	0.9498	823.3924	0.8320	21.2118	0.5001	0.0693
Rs / Total cost	0.9989 / 5922.9		0.9845 / 140.4		0.9006 / 1.0465	

The findings were obtained using genetic algorithm optimization as seen in table (6.4). The findings revealed that all the elements of a complex structure are included in the distribution, and each relies on its machine role. And we got the highest reliability of the system using exponential behavior with feasibility factor model, where Rs=0.9989, and at a cost (5922). The highest assignment to the second component R_2 = 0.9535 followed when using exponential behavior model The system was reliable Rs=9845 And at a total cost (140), and your highest assignment was for the second component R_2 = 0.99. Finally logarithmic model where was the reliability of the system Rs=0.9006. And the total cost was (1.0465) The highest allocation obtained by the second component R_2=0.9329.

6.7 Costing and Customization Using Particle Swarm

We used particle swarm optimization and three cost functions (exponential behavior with feasibility factor model, exponential behavior model, and logarithmic model) to achieve the highest system reliability at the lowest possible cost, as well as to compare the results of the three cost functions using PSO in terms of reliability allocation and optimization, and accurate reliability[1]. As shown in the table below.

Table 6.5: Summary table for a comparative results exponential behavior with feasibility factor model, exponential behavior model and logarithmic model functions by using PSO.

Components	Feasibility factor model	Cost.	Exponential behavior model	Cost.	Logarithmic model	Cost.
R_1	0.7975	2.5276	0.7994	8.8814	0.671	0.1112
R_2	0.8437	4.0942	0.8556	50.9122	0.8961	0.2264
R_3	0.7812	2.2435	0.6808	1.6769	0.7368	0.1335
R_4	0.5145	1.0185	0.5191	0.6498	0.5138	0.0721
R_5	0.5452	1.0629	0.6021	0.9601	0.7413	0.1352
R_6	0.7923	2.4281	0.6530	1.3379	0.6321	0.1000
R_7	0.6523	1.3107	0.7393	3.1571	0.6858	0.1158
R_8	0.7162	1.6060	0.7558	3.9865	0.6742	0.1121
R_9	0.7960	2.4979	0.8119	11.9664	0.7666	0.1455
R_{10}	0.6884	1.4547	0.7741	5.3735	0.7845	0.1535
Rs / Total cost	0.9019	20.2442	0.9012	88.9018	0.9001	1.3053

The findings were obtained using particle swarm optimization as seen in table (6.5). The findings revealed that all the elements of a complex structure are included in the distribution, and each relies on its machine role. And we got the highest reliability of the system using exponential behavior with feasibility factor model, where Rs=0.9019 and at a cost (20.2442) . The highest assignment to the second component R_2 = 0.8437 followed when using exponential behavior model the system was reliable Rs=0.9012, and at a total cost (88.9018) .And your highest assignment was for the second component R_2 = 0.8556. Finally logarithmic model where was the reliability of the system Rs=0.9001, and the total cost was (1.3053) The highest allocation obtained by the second component R_2=0.8961.

6.8 Costing and Customization Using Ant Colony

We used ant colony algorithm and three cost functions (exponential behavior with feasibility factor model, exponential behavior model and logarithmic model) to obtain the highest reliability of the system with the lowest possible cost and to compare the results between the three cost functions by using ACO in terms of reliability allocation and optimization, accurate reliability. As the following table.

Table 6.6: Summary table for a comparative results exponential behavior with feasibility factor model, exponential behavior model and logarithmic model functions by using ACO.

Components	Feasibility factor model	Cost.	Exponential behavior model	Cost.	Logarithmic model	Cost.
R_1	0.794	6.871	0.786	10.425	0.579	0.866
R_2	0.844	15.129	0.843	24.29	0.998	6.490
R_3	0.782	6.072	0.794	11.442	0.627	0.987
R_4	0.815	9.08	0.811	14.12	0.746	1.372
R_5	0.644	2.475	0.664	4.438	0.896	2.264
R_6	0.801	7.565	0.792	11.115	0.941	2.836
R_7	0.694	3.123	0.689	4.995	0.896	2.264
R_8	0.801	7.565	0.777	9.417	0.553	0.805
R_9	0.815	9.08	0.811	14.124	0.941	2.836
R_{10}	0.815	9.08	0.811	14.124	0.941	2.836
Rs / Total cost	0.95 / 38.023		0.947 / 59.248		0.998 / 11.781	

The findings were obtained using ant colony algorithm optimization as seen in table (6.6). The findings revealed that all the elements of a complex structure are included in the distribution, and each relies on its machine role. And we got the highest reliability of the system using logarithmic model, where Rs=0.998 and at a cost (11.781) . The highest assignment to the second component R_2 = 0.998. Followed when using exponential behavior with feasibility factor model the system was reliable Rs=0.95, and at a total cost (38.023) .And your highest assignment was for the second component R_2 = 0.844. Finally exponential behavior model where was the reliability of the system Rs=0.947 , and the total cost was (59.248) the highest allocation obtained by the second component R_2=0.843.

6.9 Costing and Customization Using Bee Colony

We used bee colony optimization and three cost functions (exponential behavior with feasibility factor model, exponential behavior model, and logarithmic model) to achieve the highest system reliability at the lowest possible cost and to compare the results between the three cost functions using ACO in terms of reliability allocation and optimization, as well as accurate reliability[3]. As shown in the table below.

Table 6.7: Summary table for a comparative results exponential behavior with feasibility factor model, exponential behavior model and logarithmic model functions by using BCO

Components	Feasibility factor model	Cost.	Exponential behavior model	Cost.	Logarithmic model	Cost.
R_1	0.809	8.312	0.795	11.462	0.732	1.316
R_2	0.858	20.514	0.853	30.005	0.999	6.907
R_3	0.822	10.064	0.779	9.606	0.604	0.926
R_4	0.809	8.312	0.795	11.462	0.522	0.738
R_5	0.766	5.138	0.674	4.635	0.954	3.079
R_6	0.822	10.064	0.776	9.319	0.954	3.079
R_7	0.744	4.276	0.713	5.709	0.826	1.748
R_8	0.822	10.064	0.795	11.462	0.55	0.798
R_9	0.822	10.064	0.795	11.462	0.954	3.079
R_{10}	0.822	10.064	0.795	11.462	0.954	3.079

Rs / Total cost	0.9616 / 48.438	0.946 / 58.293	0.999 / 12.376

The findings were obtained using bees colony optimization as seen in table (6.7). The findings revealed that all the elements of a complex structure are included in the distribution, and each relies on its machine role. And we got the highest reliability of the system using logarithmic model, where Rs=0.999 and at a cost (12.376) . The highest assignment to the second component R_2 = 0.999. Followed when using exponential behavior with feasibility factor model The system was reliable Rs=0.9616, and at a total cost (48.438) .And your highest assignment was for the second component R_2 = 0.858. Finally exponential behavior model where was the reliability of the system Rs=0.946, and the total cost was (58.293) The highest allocation obtained by the second component R_2=0.853.

6.10 Calculating Reliability Importance of GA

The reliability importance of each component of the system was calculated after solving the allocation problem. We will calculate the significance using the significance equation (4.20) and apply it to the assignment values of the genetic algorithm with the three cost functions as in the following table.6.8.

Table 6.8: Summary table for reliability importance of GA of given system

Components	Feasibility factor model	Exponential behavior model	Logarithmic model
R_1	0.006	0.014	0.023
R_2	0.021	0.219	0.714
R_3	0.006	0.013	0.023
R_4	0.006	0.014	0.023
R_5	0.003	0.016	0.026
R_6	0.003	0.07	0.441
R_7	0.003	0.016	0.026
R_8	0.006	0.013	0.023
R_9	0.009	0.048	0.058
R_{10}	0.009	0.046	0.058

The results by using cost function (feasibility factor model) were as follows: The value of Better importance was given to compound R_2 (0.21). The results by using cost function (the exponential behavior model). The value of Better importance was given to compound R_2 (0.219) and the results by using cost function (the Logarithmic model) The value of Better importance was given to compound R_2 (0.714).

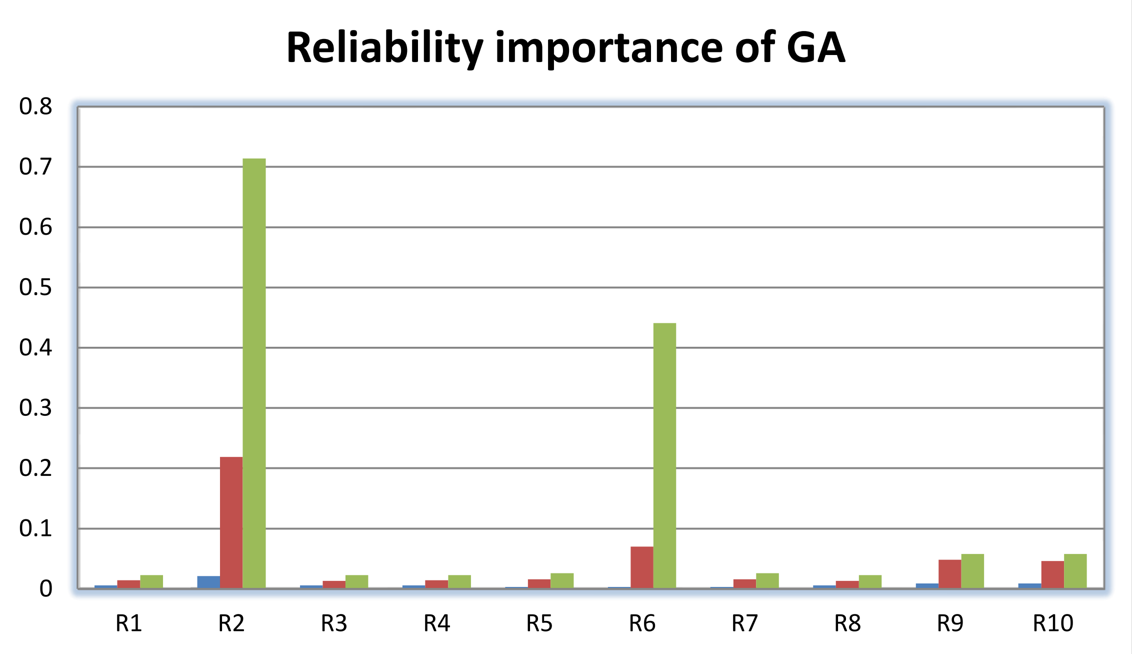


Figure 6.4: Reliability importance of GA for the three cost functions

6.11 Calculating Reliability Importance of PSO

After solving the allocation of reliability values for each component of the system by reparation them in eq. (4.20), derivatives where the results of the importance of reliability apply it to the assignment values of the particle swarm optimization with the three cost functions as in the following table.6.9.

Table 6.9:Summary table for reliability importance of PSO of given system

Components	Feasibility factor model	Exponential behavior model	Logarithmic model
R_1	0.0530	0.0527	0.0364
R_2	0.3779	0.3566	0.4036
R_3	0.0644	0.0731	0.0603
R_4	0.0821	0.0812	0.0475
R_5	0.0343	0.0457	0.0562
R_6	0.1469	0.1045	0.1106
R_7	0.0222	0.0421	0.0473
R_8	0.0703	0.0658	0.0659
R_9	0.0929	0.1098	0.1148
R_{10}	0.1208	0.1471	0.1388

The results by using cost function (feasibility factor model) were as follows: The value of Better importance was given to compound R_2 (0.3779). The results by using cost function (the exponential behavior model). The value of Better importance was given to compound R_2 (0.3566) and the results by using cost function (the Logarithmic model) The value of Better importance was given to compound R_2 (0.4036).

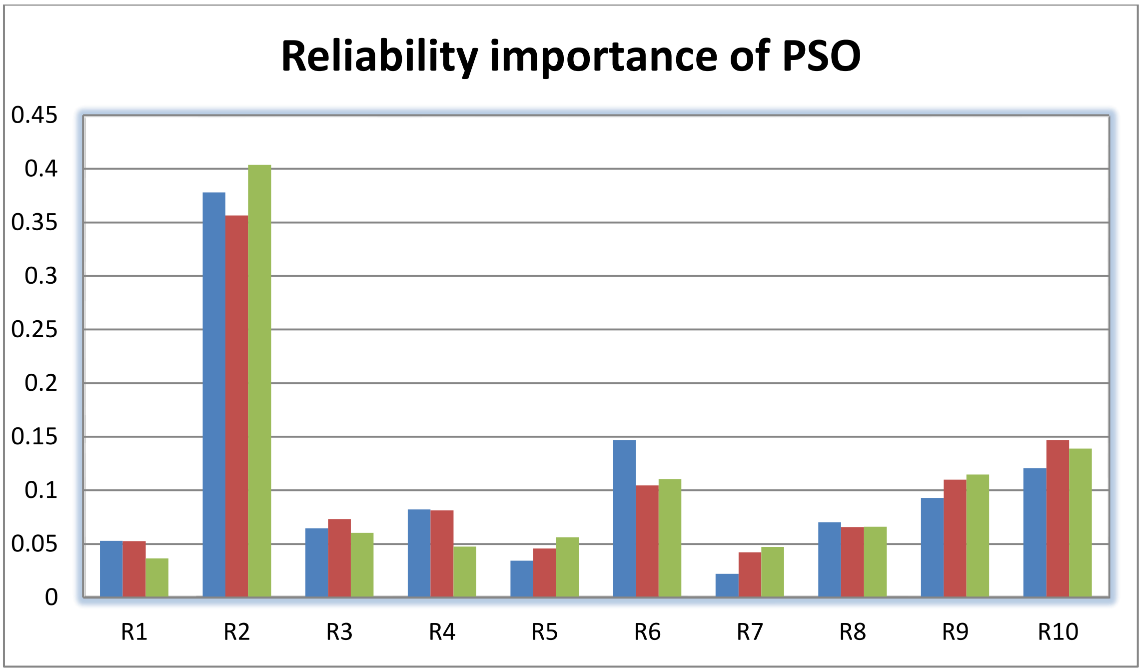


Figure 6.5: Reliability importance of PSO for the three cost functions

6.12 Calculating Reliability Importance of ACO

The reliability importance of each component of the system was calculated after solving the allocation problem We will calculate the significance using the significance equation (4.20) and apply it to the assignment values of the ant colony algorithm with the three cost functions as in the following table.6.10

Table 6.10:Summary table for reliability importance of ACO of given system

Components	Feasibility factor model	Exponential behavior model	Logarithmic model
R_1	0.0618	0.0627	0.0015
R_2	0.2154	0.2243	0.3991
R_3	0.0578	0.0577	0.001
R_4	0.0602	0.0607	0.0011
R_5	0.0134	0.0147	0.0039
R_6	0.0675	0.0693	0.0141
R_7	0.0147	0.0157	0.0041
R_8	0.0564	0.0589	0.0011
R_9	0.0903	0.0940	0.0011
R_{10}	0.0889	0.0912	0.0067

The results by using cost function (feasibility factor model) were as follows: The value of Better importance was given to compound R_2 (0.2154). The results by using cost function (the exponential behavior model). The value of Better importance was given to compound R_2 (0.2243) and the results by using cost function (the Logarithmic model) The value of Better importance was given to compound R_2 (0.3991).

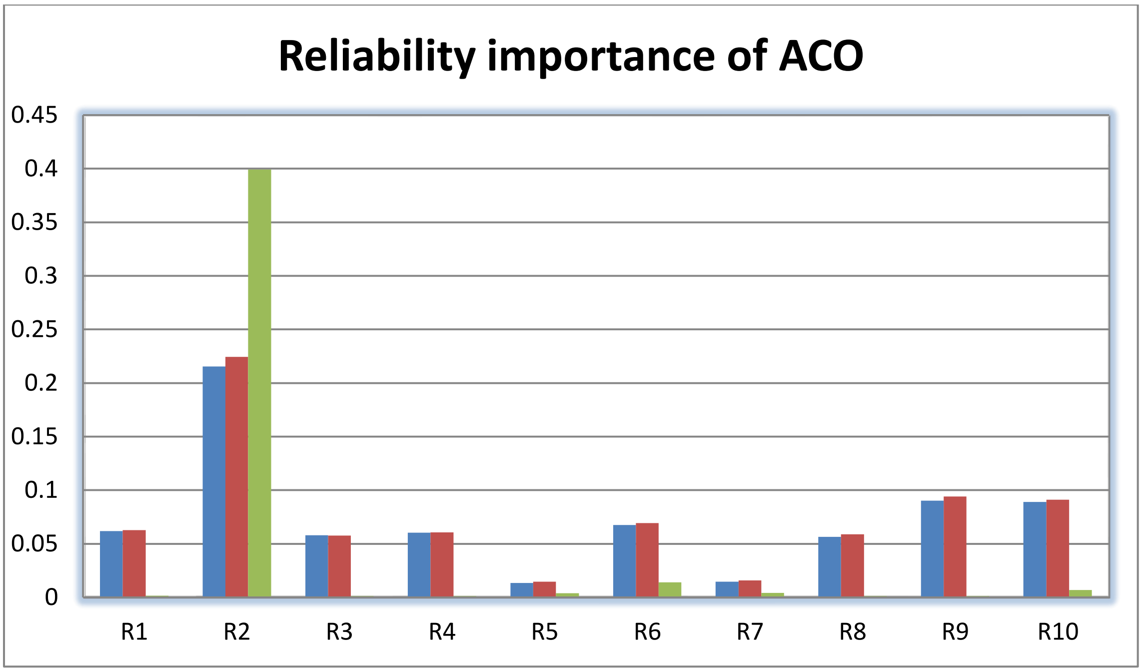


Figure 6.6: Reliability importance of ACO for the three cost functions

6.13 Calculating Reliability Importance of BCO

After solving the allocation problem, the dependability importance of each component of the system was computed. We'll use the significance equation (4.20) to determine the significance and apply it to the assignment values of the bee colony optimization with the three cost functions in the table below. 6.11

Table 6.11:Summary table for reliability importance of BCO of given system

Components	Feasibility factor model	Exponential behavior model	Logarithmic model
R_1	0.0478	0.0587	0.09
R_2	0.1947	0.2329	0.4336
R_3	0.0508	0.0566	0.117
R_4	0.0478	0.0587	0.1297
R_5	0.0107	0.0163	0.0042
R_6	0.0516	0.0748	0.0115
R_7	0.0100	0.0174	0.0021

R_8	0.0508	0.0555	0.1287
R_9	0.0759	0.0962	0.0073
R_{10}	0.0786	0.0964	0.0032

The results by using cost function (feasibility factor model) were as follows: The value of Better importance was given to compound R_2 (0.1947). The results by using cost function (the exponential behavior model). The value of Better importance was given to compound R_2 (0.2329) and the results by using cost function (the Logarithmic model) The value of Better importance was given to compound R_2 (0.4336).

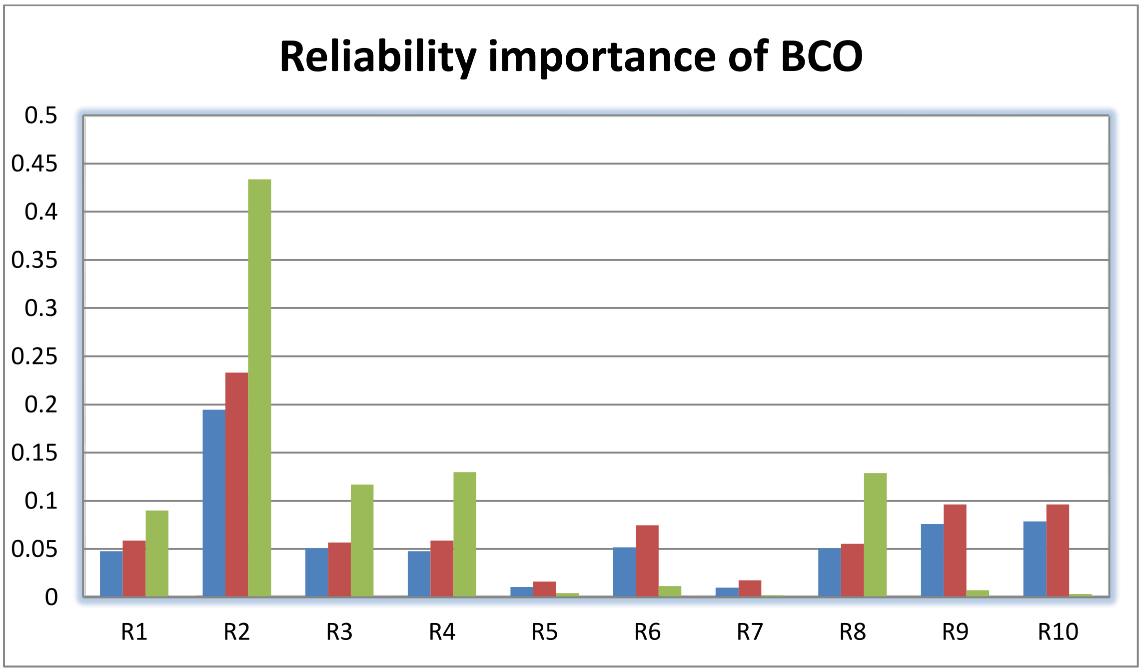


Figure 6.7: Reliability importance of BCO for the three cost functions

Chapter Seven

7.1 Conclusion

The best allocation of reliability was calculated using algorithms: (1) genetic algorithm (A 4), (2) particle swarm optimization (A 5), bees colony optimization (A 6) and ant colony algorithm (A 7). The three cost functions were used where the allocation was calculated for each component of the system. Then, we calculated the cost for each component and the total cost of the system also, calculating the exact reliability as shown in the tables (6.1), (6.2),(6.3), (6.4), (6.7). Finally, we have calculated the importance of each component of the system as in the tables (6.8), (6.9),(6.10), (6.11).

7.2 Future Works

1. The possibility of studying the optimization reliability allocation of electric generator inside aircraft system by (genetic algorithm ,particle swarm optimization, ant colony algorithm and bees colony optimization) as shown in Fig.(7.1)

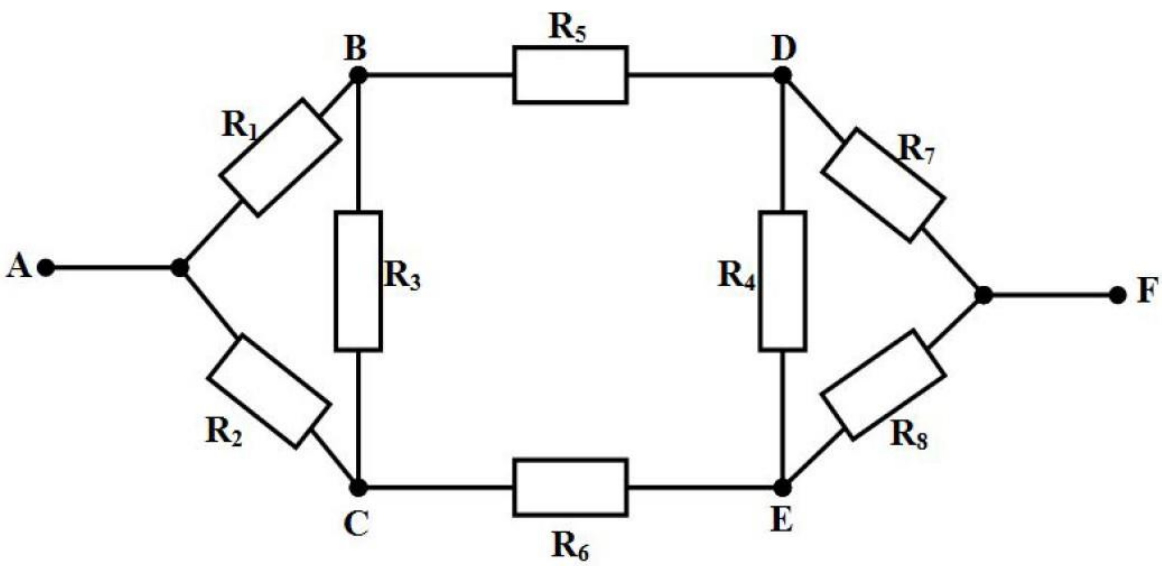


Figure 7.1 Electric generator inside aircraft system

2. The study of domination in graphs with the application of networks reliability.

Appendix

Appendix

In this appendix, we will review some of the key algorithms, programs and features used in multivariate reliability polynomial calculations and piloting and calculate the reliability optimization allocation based on the mathematical constructor studied in this thesis. Most of the Matlab∗ software used algorithms, programs and features.

A 1. Graphs Conception

This program is used to draw the graph contained in the thesis yEd Graph Editor

A 2.Generation of minimal paths Algorithm

This algorithm is based on what we have studied in the third chapter to Generation of minimal paths of graph g, where depends on several functions in Matlab to give the exact result in general.

```
syms xi
CN={Connection Matrix};
nn=size(CN,1);
for n=nn:-1:3

    for i=1:n-1
        for j=3:n
            if i~=j && i~=2
                P=cell2mat(CN{i,j});
                if P=='0'
                     CN1(i,j)={strcat('(', CN{i,2},'*',
CN{2,j},')')};
                else
                   CN1(i,j)={strcat('(',CN{i,j} ,'+',
CN{i,2},'*', CN{2,j},')')};
                end
            else
                CN1(i,j)={'1'};
            end
        end
    end
    CN1(2,:) =  [];
    CN1(:,2) = [];
    CN=CN1;
end
pat=char(CN{1,2});
eval(strcat('result=expand',pat,';'));
C = char(strsplit(char(result),'+'));
j=0;
for i=1:size(C,1)
    T=strfind(C(i,:),'^');
    if isempty(T)
        j=j+1;
        CC(j,:)=C(i,:);

    end

end
disp(CC);
```

A 3.Generation of minimal cuts Algorithm

This algorithm is based on what we studied in the third chapter to generate minimal graph g cuts, where the precise outcome in general depends on multiple features in Matlab.

```
IM=[matrix]
[m,n]=size(IM);

mc=zeros(n,n);
for i=2:n
    c = combnk(1:n,i);
    k1=1;
    for j=1:size(c,1)

        cut=zeros(1,1);
        B=c(j,1:end);
        ok=0;
        for k=1:m
             if sum(IM(k,B))>0
                ok=ok+1;
             end
        end

    if ok==m
        stoop=true;
        for kk=1:length(B)-1
             cc = combnk(B,kk);
             for jj=1:size(cc,1)
                 ok1=0;
                 for t=1:m
                          if sum(IM(t,cc(jj,1:end)))>0
                              ok1=ok1+1;
                           end
                 end
                  if ok1>=m
                      stoop=false;
                  end
              end
         end

        if  stoop
              cutA(i-1).cut(k1,1:size(c,2))=c(j,:);
              k1=k1+1;
              st=cutA(i-1).cut;
        end
    end
    end
end
 cutA.cut
```

A4. Genetic Algorithm

This algorithm is based on what we have studied in the chapter six reliability allocation and optimization, where depends on Matlab to give the exact result in general.

```
function [c,ceq]=constraint(R)
Rg<=Rs;
      c=[Rg-Rs];
Rs;
ceq=[];
c;
function y=objectivefun(R,ai,bi)
function

 cost function;
end
Minprogram;
ai=input('ai=');
bi=input('bi=');
llb=[0.5]
uub=[0.99];
 nvars=i;
objfun=@(R)objectivefun(R,ai);
 constfunc=@constraint;
[R,faval]=ga(objfun,nvars,[],[],[],[],llb,uub,constfunc)
```

A5. Particle Swarm Optimization

This algorithm is based on what we have studied in the chapter six reliability allocation and optimization, where depends on Matlab to give the exact result in general.

```
function
[Swarm,Best,G,OF,RS,TF]=F_PSO_Fitness(NumSwarms,Swarm,Nkey,Ind,B
est,G,URs,a,b,f)
RG = <Rs; TF=0;
for H = 1:NumSwarms
        R=Swarm{1,1}(H,:);
    [C,Ci,Rs]=F_Objective(R,Nkey,a,b,f);
    OF(H)=C; RS(H)=Rs;
    Swarm{1,2}(H,:)=Ci; Swarm{2,1}(H)=C; Swarm{3,3}(H)=Rs;
    FitH=Swarm{2,3}(H); FitG=Swarm{3,3}(H);

    if (Swarm{3,3}(H) <= Swarm{2,3}(H))
   end;
    FitH=Swarm{3,3}(H); FitG=Swarm{3,3}(G);

    if Swarm{3,3}(H) <= Swarm{3,3}(G)
        Swarm{1,1}(G,:) = Swarm{1,1}(H,:);
        Swarm{2,1}(G) = Swarm{2,1}(H);
        Swarm{3,3}(G) = Swarm{3,3}(H);
        G = H;
    end
```

```
function
[Swarm,Best]=F_PSO_Initialize(Nkey,NumSwarms,Vmax,lb,ub)
IntUB=10^10; Vmin=-Vmax;
for I=1:NumSwarms
    Position = ilb + (ub-ilb).*rand(Nkey,1);
    Swarm{1,1}(I,:) = Position;
    Swarm{1,2}(I,:) = lb;
    Swarm{1,3}(I,:) = zeros(1,Nkey);
    Swarm{2,1}(I) = 2*IntUB;
    Swarm{2,3}(I) = 2*IntUB;
Swarm{3,1}(I) = 2*IntUB;
Swarm{3,2}(I,:)= Vmin + (Vmax Vmin).*rand(Nkey,1);
    Swarm{3,3}(I) = 2*IntUB;
end
Swarm{2,2}=1;
Best{1,1}=[];  Best{1,2}=[];  Best{2,1}=2*IntUB;
Best{2,2}=2*IntUB;

function [C,Ci,Rs]=F_Objective(R,Nkey,a,b,f)

Rs;
C=0;
for i=1:Nkey
    cost function
C=C+Ci(i);
end
end
```

```
function
[Swarm]=F_PSO_Evolution(Nkey,NumSwarms,Swarm,G,Vmax,ALPH,lb)
Vmin=-Vmax;
C1 = 1; C2 = 1;
for I = 1 : NumSwarms
    for J = 1 : Nkey
        PD = Swarm{1,3}(I,J)-Swarm{1,1}(I,J);
        ND = Swarm{1,3}(G,J)-Swarm{1,1}(I,J);
        del1 = C1*rand(1,1) * PD + C2*rand(1,1) * ND;
        del = del1 + ALPH * Swarm{3,2}(I,J);
        del = Swarm{1,1}(I,J) + del;
        if del > Vmax
            del = del-Vmax;
        end;
        if del < Vmin
            del = del+Vmax;
        end;
        Swarm{3,2}(I,J)= del;
        Y = Swarm{1,1}(I,J)+Swarm{3,2}(I,J);

        end
    end

    R=Z;
    Swarm{1,1}(I,:)=R;
end;
```

```
MaxIter=10000;
Exm = 1; NumSwarms=20;
fprintf(' No of iterations= %d\n',MaxIter);
Nkey = i;
ub=0.99;
lb=0.5;
a(1:Nkey)=0.1; b(1:Nkey)=0.9; f(1:Nkey)=0.4;

URs = 1.0;
fprintf(' Upper Bound of Rs = %3.2f\n',URs);
Ind=1; G=1; Vmax=1; JJ=0;
tic;
Swarms=[]; Best=[]; Ind=1;
[Swarms,Best]=F_PSO_Initialize(Nkey,NumSwarms,Vmax,lb,ub);
for Ind = 1 : MaxIter
    ALPH = 0.4 + 0.5*((MaxIter-Ind)/MaxIter);
Swarms=F_PSO_Evolution(Nkey,NumSwarms,Swarms,G,Vmax,ALPH,lb);
    if TF==1
        JJ=JJ+1;
        RR(:,JJ)=Best{1,2};
        Cost(JJ)=Best{2,1};
    end;
end;
fprintf(' The Values ( Ri , Ci(Ri) )\n');
for i=1:Nkey
    fprintf(' R(%d)=  ',i);
    for j=1:JJ
        cost function;
        fprintf('(%5.4f,%3.2f),',RR(i,j),Ci(i));
    end;
    fprintf('\n');
end;
C=sum(Ci);
fprintf(' Total Cost (C)= %5.4f\n',C);
fprintf(' Best Rs = %5.4f\n',Best{2,2});
fprintf(' End of Program.... \n');
```

A6. Bees colony optimization

This algorithm is based on what we have studied in the chapter six reliability allocation and optimization, where depends on Matlab to give the exact result in general.

```
clc
clearvars
rng('default')
Number_caindedate=input('Number of candetate solution=');
cost_function=input('pleas input cost function Number \in [1,3]: \n 1=EXP. with faciblty factor\n 2=EXP\n 3=LOG. \n cost function Number=');
cf=cost_function;
tic
Rg=0.9;
No_cand=Number_caindedate;
RG=Rg;
COLN=No_cand;
nep=3;
RiS_COLN=zeros(11,COLN);
Costi_COLN=RiS_COLN;
rau=0.500:0.001:0.999;
n=length(rau);
m=ceil(n/4);
N=n-m;
Rau=ones(10,n);
New_Rau=Rau;
Rm=Rau;
sol=Rau;
%R=ones(10,1);
S=ones(n,1);
ci=Rau;
cost=S';
RS=S'*0;
BEST_COST=ones(1,COLN)*inf;
BEST_Ci=ones(10,COLN)*inf;
BEST_Rs=BEST_COST;
BEST_Ri=BEST_Ci;
MIN_COST=inf;
rr=ones(10);
for i=1:1000
   rr(:,i)=randperm(10)';
end
for i=1:10
   Rau(i,:)=rau;
end
for i=1:10
   for j=1:n
      r=randi([10 n],1,1);
      sol(i,j)=r;
      Rm(i,j)=rau(r);
   end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
colony=1;
for j=1:n
    RS(j)=Rsf(Rm(:,j));
    if RS(j)< RG
        state=0;
        while state==0
            rss =randsample(rau,10,true, rau );
            RS(j)=Rsf(rss');
            if RS(j)< RG
                state=0;
            else
                state=1;
                Rm(:,j)=rss;
            end
        end
    end
    if cf==1
        [ci(:,j),cost(j)]=SumCi_Rfi(Rm(:,j),0.5,0.5);
    elseif cf==2
        [ci(:,j),cost(j)]=SumCi_EXP(Rm(:,j),0.5,0.5);
    elseif cf==3
        [ci(:,j),cost(j)]=SumCi_LOG(Rm(:,j),0.5);
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[cost1,ind]=sort(cost);
pop1=Rm(:,ind);
RS=RS(ind);
pop=[pop1;RS];
cand_sol=pop(:,1:COLN);
C2=ci(:,ind);
C2=[C2;cost1];
cand_cost=C2(:,1:COLN);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:COLN
    tst=cand_sol(1:10,i);
    for j=1:nep
        ii=randi(100);
        r=rr(:,ii);
        tm=tst(r);
        rs=Rsf(tm);
        if rs > RG
            cand_sol(:,i)=[tm;rs];
            RS(i)=rs;
            if cf==1
                [cia,costa]=SumCi_Rfi(tm,0.5,0.5);
            elseif cf==2
                [cia,costa]=SumCi_EXP(tm,0.5,0.5);
            elseif cf==3
                [cia,costa]=SumCi_LOG(tm,0.5);
            end
            cand_cost(1:11,i)=[cia;costa];
        end
    end
```

```
end
[cand_cost(11,:),ind2]=sort(cand_cost(11,:));
CS=cand_cost(1:10,:);
cand_cost(1:10,:)=CS(:,ind2);
cand_sol=cand_sol(:,ind2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Costi_COLN(:,colony)=cand_cost(:,1);
RiS_COLN(:,colony)=cand_sol(:,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if cand_cost(11,1)<BEST_COST(colony)
   BEST_COST(colony)=cand_cost(11,1);
   BEST_Ci(:,colony)=cand_cost(1:10,1);
   BEST_Ri(:,colony)= cand_sol(1:10,1);
   BEST_Rs(colony)=cand_sol(11,1);
   %js=j;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if MIN_COST> BEST_COST(colony)
   MIN_COST=BEST_COST(colony);
   Ri=[BEST_Ri(:,colony);BEST_Rs(colony)];
   Ci=[BEST_Ci(:,colony);BEST_COST(colony)];
end
AC_time=toc;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for colony=2:COLN
   tic
   CS1=cand_cost(11,:);
   SCS=sum(CS1);
   pp=CS1/SCS;
   for Rii=1:10
      cs=cand_sol(Rii,:);
      New_Rau(Rii,1:N)=randsample(cs,N,true, pp );
   end
   for j=N+1:n
      for i=1:10
         r=randi([50 n],1,1);
         New_Rau(i,j)=rau(r);
      end
   end
   Rm=New_Rau;
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   for j=1:n
      RS(j)=Rsf(Rm(:,j));
      if RS(j)< RG
         state=0;
         while state==0
            rss = 0.501 + (0.999-0.501)*rand(10,1);
            RS(j)=Rsf(rss);
            if RS(j)< RG
               state=0;
            else
               state=1;
               Rm(:,j)=rss;
            end
         end
```

```
    end
    if cf==1
      [ci(:,j),cost(j)]=SumCi_Rfi(Rm(:,j),0.5,0.5);
    elseif cf==2
      [ci(:,j),cost(j)]=SumCi_EXP(Rm(:,j),0.5,0.5);
    elseif cf==3
      [ci(:,j),cost(j)]=SumCi_LOG(Rm(:,j),0.5);
    end
  end
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  [cost1,ind]=sort(cost);
  pop1=Rm(:,ind);
  RS=RS(ind);
  pop=[pop1;RS];
  cand_sol=pop(:,1:COLN);
  C=[ci;cost];
  C2=C(:,ind);
  cand_cost=C2(:,1:COLN);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  for i=1:COLN
    tst=cand_sol(1:10,i);
    for j=1:nep
      ii=randi(100);
      r=rr(:,ii);
      tm=tst(r);
      rs=Rsf(tm);
      if rs > RG
        cand_sol(:,i)=[tm;rs];
        RS(i)=rs;
        if cf==1
          [cia,costa]=SumCi_Rfi(tm,0.5,0.5);
        elseif cf==2
          [cia,costa]=SumCi_EXP(tm,0.5,0.5);
        elseif cf==3
          [cia,costa]=SumCi_LOG(tm,0.5);
        end
        cand_cost(1:11,i)=[cia;costa];
      end
    end
  end
  [cand_cost(11,:),ind2]=sort(cand_cost(11,:));
  CS=cand_cost(1:10,:);
  cand_cost(1:10,:)=CS(:,ind2);
  cand_sol=cand_sol(:,ind2);
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  Costi_COLN(:,colony)=cand_cost(:,1);
  RiS_COLN(:,colony)=cand_sol(:,1);
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  if cand_cost(11,1)<BEST_COST(colony)
    BEST_COST(colony)=cand_cost(11,1);
    BEST_Ci(:,colony)=cand_cost(1:10,1);
    BEST_Ri(:,colony)= cand_sol(1:10,1);
    BEST_Rs(colony)=cand_sol(11,1);
    %js=j;
```

```
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if MIN_COST> BEST_COST(colony)
        MIN_COST=BEST_COST(colony);
        Ri=[BEST_Ri(:,colony);BEST_Rs(colony)];
        Ci=[BEST_Ci(:,colony);BEST_COST(colony)];
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    AT=toc;
    AC_time=AC_time+AT;
    if AC_time>600
        break;
    end
end
b1=figure;
bar((Ri(1:10)'));
hold on
bar(11,Ri(11),'r')
legend('Ri','Rs','location','north')
if cf==1
    title(['Ri and Rs: function: faciblty Exp with BEE colony Number=',num2str(No_cand)]);
    pname=['Ri and Rs function is faciblty Exp with BEE colony Number=',num2str(No_cand)];
elseif cf==2
    title(['Ri and Rs: function: Exponoatial with BEE colony Number=',num2str(No_cand)]);
    pname=['Ri and Rs function is Exponoatial with BEE colony Number=',num2str(No_cand) ];
elseif cf==3
    title(['Ri and Rs: function: Logarithm with BEE colony Number=',num2str(No_cand)]);
    pname=['Ri and Rs function is Logarithm with BEE colony Number=',num2str(No_cand) ];
end
print(pname,'-dpng')
f2=figure;
bar(Ci(1:10)');
hold on
bar(11,Ci(11),'r');
legend('Ci','Cost','location','northwest')
if cf==1
    title(['Ci and cost function: faciblty Exp with BEE colony Number=',num2str(No_cand)]);
    pname=['Ci and cost function is faciblty Exp with BEE colony Number=',num2str(No_cand)
];
elseif cf==2
    title(['Ci and cost function: Exponoatial with BEE colony Number=',num2str(No_cand)]);
    pname=['Ci and cost function is Exponoatial with BEE colony Number=',num2str(No_cand)
];
elseif cf==3
    title(['Ci and cost function: Logarithm with BEE colony Number=',num2str(No_cand)]);
    pname=['Ci and cost function is Logarithm with BEE colony Number=',num2str(No_cand) ];
end
print(pname,'-dpng')
if cf==1
    filename=['BEE_COLONY_(EXP_Fi)_(COLONY)=',num2str(No_cand),'.xlsx'];
elseif cf==2
    filename=['BEE_COLONY_(EXP)_(COLONY)=',num2str(No_cand),'.xlsx'];
elseif cf==3
    filename=['BEE_COLONY_(LOG)_(COLONY)=',num2str(No_cand),'.xlsx'];
end
```

A7. Ant colony algorithm

This algorithm is based on what we have studied in the chapter six reliability allocation and optimization, where depends on Matlab to give the exact result in general.

```
clc

clearvars
clf reset
rng('default')
tic
Rg=0.9;
No_colony=input('Number of Colony=');
cost_function=input('pleas input cost function Number \in [1,3]: \n 1=EXP. with faciblty
factor\n 2=EXP\n 3=LOG. \n cost function Number=');
cf=cost_function;
s=rng;
%rng(s);
RG=Rg;
COLN=No_colony;
RiS_COLN=zeros(11,COLN);
Costis_COLN=RiS_COLN;
rau=0.500:0.001:0.999;
n=length(rau);
Rau=ones(10,n);
New_Rau=Rau;
tau=Rau;
Rm=Rau;
sst=Rau;
sol=Rau;
R=ones(10,1);
S=ones(n,1);
ci=Rau;
cost=S';
RS=S;

BEST_COST=ones(1,COLN)*inf;
BEST_Ci=ones(10,COLN)*inf;
BEST_Rs=BEST_COST;
BEST_Ri=BEST_Ci;
MIN_COST=inf;
for i=1:10
   Rau(i,:)=rau;
end
for i=1:10
   for j=1:n
       r=randi([10 n],1,1);
       sol(i,j)=r;
       Rm(i,j)=rau(r);
   end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
colony=1;
for j=1:n
   RS(j)=Rsf(Rm(:,j));
   if RS(j)< RG
      state=0;
      while state==0
         rss = 0.501 + (0.999-0.501)*rand(10,1);
         RS(j)=Rsf(rss);
         if RS(j) < RG
            state=0;
         else
            state=1;
            Rm(:,j)=rss;
         end
      end
   end

   if cf==1
      [ci(:,j),cost(j)]=SumCi_Rfi(Rm(:,j),0.5,0.5);
   elseif cf==2
      [ci(:,j),cost(j)]=SumCi_EXP(Rm(:,j),0.5,0.5);
   elseif cf==3
      [ci(:,j),cost(j)]=SumCi_LOG(Rm(:,j),0.5);
   end

   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   if  cost(j) < BEST_COST(colony)
      BEST_COST(colony)=cost(j);
      BEST_Rs(colony)=RS(j);
      BEST_Ci(:,colony)=ci(:,j);
      BEST_Ri(:,colony)=Rm(:,j);
      js=j;
      RiS_COLN(:,colony)=[BEST_Ri(:,colony);BEST_Rs(colony)];
      Costis_COLN(:,colony)=[BEST_Ci(:,colony);BEST_COST(colony)];
   end
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
if BEST_COST(colony)< MIN_COST
   MIN_COST=BEST_COST(colony);
   Ri=RiS_COLN(:,colony);
   Ci=Costis_COLN(:,colony);
   km=colony;
end
wcost=1./cost;
for i=1:10
   for j=1:n
      tau(i,j)=tau(i,j)+(wcost(j)).*tau(i,j);
   end
end
tau(:,js)=2*tau(:,js)*MIN_COST;
tau(tau<0)=0;
AC_time=toc;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for colony=2:COLN
   tic
```

```
%        stts=sum(tau,2);
%        for c=1:n
%            sst(:,c)=stts;
%        end
%        pp=tau./(sst-tau);
    for Rii=1:10
        New_Rau(Rii,:)=randsample( Rm(Rii,:), n, true, tau(Rii,:) );
    end
%     for i=1:10
%        for j=1:n
%            r=randi([10 n],1,1);
%            sol(i,j)=r;
%            Rm(i,j)=New_Rau(r);
%        end
%     end
Rm=New_Rau;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for j=1:n
        RS(j)=Rsf(Rm(:,j));
        if RS(j)< RG
            state=0;
            while state==0
                rss = 0.501 + (0.999-0.501)*rand(10,1);
                RS(j)=Rsf(rss);
                if RS(j) < RG
                    state=0;
                else
                    state=1;
                    Rm(:,j)=rss;
                end
            end
        end
    if cf==1
            [ci(:,j),cost(j)]=SumCi_Rfi(Rm(:,j),0.5,0.5);
        elseif cf==2
            [ci(:,j),cost(j)]=SumCi_EXP(Rm(:,j),0.5,0.5);
        elseif cf==3
            [ci(:,j),cost(j)]=SumCi_LOG(Rm(:,j),0.5);
        end
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        if  cost(j) < BEST_COST(colony)
            BEST_COST(colony)=cost(j);
            BEST_Rs(colony)=RS(j);
            BEST_Ci(:,colony)=ci(:,j);
            BEST_Ri(:,colony)=Rm(:,j);
            js=j;
            RiS_COLN(:,colony)=[BEST_Ri(:,colony);BEST_Rs(colony)];
            Costis_COLN(:,colony)=[BEST_Ci(:,colony);BEST_COST(colony)];
        end
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    end
    if BEST_COST(colony)< MIN_COST
        MIN_COST=BEST_COST(colony);
        Ri=RiS_COLN(:,colony);
        Ci=Costis_COLN(:,colony);
        km=colony;
```

```
    end
    wcost=1./cost;
    for i=1:10
        for j=1:n
            tau(i,j)=tau(i,j)+(wcost(j)).*tau(i,j);
        end
    end
    tau(:,js)=2*tau(:,js)*MIN_COST;
    tau(tau<0)=0;
    AT=toc;
    AC_time=AC_time+AT;
    if AC_time>600
        break;
    end
end
b1=figure;
bar((Ri(1:10)'));
hold on
bar(11,Ri(11),'r')
legend('Ri','Rs','location','north')
if cf==1
    title(['Ri and Rs: function: faciblty Exp with ANT colony Number=',num2str(No_colony)]);
    pname=['Ri and Rs function is faciblty Exp with ANT colony
Number=',num2str(No_colony)];
elseif cf==2
    title(['Ri and Rs: function: Exponoatial with ANT colony Number=',num2str(No_colony)]);
    pname=['Ri and Rs function is Exponoatial with ANT colony Number=',num2str(No_colony)
];
elseif cf==3
    title(['Ri and Rs: function: Logarithm with ANT colony Number=',num2str(No_colony)]);
    pname=['Ri and Rs function is Logarithm with ANT colony Number=',num2str(No_colony)
];
end
print(pname,'-dpng')
f2=figure;
bar(Ci(1:10)');
hold on
bar(11,Ci(11),'r');
legend('Ci','Cost','location','northwest')
if cf==1
    title(['Ci and cost function: faciblty Exp with ANT colony Number=',num2str(No_colony)]);
    pname=['Ci and cost function is faciblty Exp with ANT colony
Number=',num2str(No_colony) ];
elseif cf==2
    title(['Ci and cost function: Exponoatial with ANT colony Number=',num2str(No_colony)]);
    pname=['Ci and cost function is Exponoatial with ANT colony
Number=',num2str(No_colony) ];
elseif cf==3
    title(['Ci and cost function: Logarithm with ANT colony Number=',num2str(No_colony)]);
    pname=['Ci and cost function is Logarithm with ANT colony Number=',num2str(No_colony)
];
end
print(pname,'-dpng')
if cf==1
    filename=['ANT_COLONY_(EXP_Fi)_(COLONY)=',num2str(No_colony),'.xlsx'];
elseif cf==2
```

```
    filename=['ANT_COLONY_(EXP)_(COLONY)=',num2str(No_colony),'.xlsx'];
elseif cf==3
    filename=['ANT_COLONY_(LOG)_(COLONY)=',num2str(No_colony),'.xlsx'];
end
```

References

[1] Abdullah, Ghazi, and Zahir Abdul Haddi Hassan. "Using of particle swarm optimization (PSO) to addressed reliability allocation of complex network." Journal of Physics: Conference Series. Vol. 1664. No. 1. IOP Publishing, 2020.

[2] Abdullah, Ghazi, and Zahir Abdul Haddi Hassan. "A Comparison Between Genetic Algorithm and Practical Swarm to Investigate the Reliability Allocation of Complex Network." Journal of Physics: Conference Series. Vol. 1818. No. 1. IOP Publishing, 2021.

[3] Abdullah, Ghazi, and Zahir Abdul Haddi Hassan. "Use of Bees Colony algorithm to allocate and improve reliability of complex network." Journal of Physics: Conference Series. Vol. 1999. No. 1. IOP Publishing, 2021.

[4] Abdullah, Ghazi, and Zahir Abdul Haddi Hassan. "Using of Genetic Algorithm to Evaluate Reliability Allocation and Optimization of Complex Network." IOP Conference Series: Materials Science and Engineering. Vol. 928. No. 4. IOP Publishing, 2020.

[5] Abed, S. A. and Udriste, C., Optimization Techniques and Methods in Reliability Allocation, Ph. D. Thesis, University Politehnica of Bucharest,(2017)

[6] Abed, S. A., Udriste, C. and Tevy, I., Optimal reliability allocation for redundancy series-parallel systems, European Journal of Pure and Applied Mathematics, 10(4), pp. 877-889,(2017),.

[7] Abed, Saad Abbas, Hatem Kareem Sulaiman, and Zahir Abdul Haddi Hassan. "Reliability Allocation and Optimization for (ROSS) of a Spacecraft by using Genetic Algorithm." Journal of Physics: Conference Series. Vol. 1294. No. 3. IOP Publishing, 2019.

[8] Abraham, J. A., An improved algorithm for network reliability. IEEE Transactions on Reliability, R-28:58-61, April (1979).

[9] Aggarwal, K.K., Reliability engineering,Center for Excellemce in

Reliability engineering, Regional engineering College ,Kurukshetra, India,(1993).

[10] AGREE . Reliability of Military Electronic Equipment. Tech. Rep. Washington, DC: Advisory Group on Reliability of Electronic Equipment, U.S. Department of Defense. (1957).

[11] Alghamdi, S. M. and Percy, D. F., Reliability equivalence factors for a series-parallel system of components with exponentiated Weibull lifetimes, IMA Journal of Management Mathematics, vol. 28.3, pp. 339-358,(2015),.

[12] Allan, R. N., Billinton, R. and De Oliveira, M. F., An efficient algorithm for deducing the minimal cuts and reliability indices of a general network configuration, IEEE Transactions on Reliability, vol. 25.4, pp. 226-233, (1976).

[13] Andrea Bobbio, Caterina Ferraris, and Roberta Terruggia. New challenges in network reliability analysis. CNIP, 6:554{564, 2006.

[14] Ansell, J. I. and Phillips M. J., Practical Methods for Reliability Data Analysis, Oxford University Press, New York, (1994).

applied probability and statistics. Methuen, London: Methuen & Co. (1962).

[15] Beichelt, F. and Tittmann, P., Reliability and maintenance: networks and systems, CRC Press, (2012).

[16] Biegel, J. E., Determination of tie sets and cut sets for a system without feedback, IEEE Transactions on Reliability, vol. 26.1, pp. 39-42, (1977).

[17] Billinton, R. and Allan, R. N., Reliability evaluation of engineering systems: concepts and techniques, Springer Science and Business Media, (2013).

[18] Chelson, P. O. and Eckstein, E., Reliability computation from reliability block diagrams, IEEE Transactions on Reliability, vol. 25, pp. 283-283, (1975).

[19] Chen, W. K., Graph theory and its engineering applications, vol. 5, World Scientific, (1997).

[20] Chithra, M. and Vijayakumar, A, The diameter variability of the Cartesian product of graphs", Discrete Math, vol. Algorithms Appl. 6 ,pp. 289 -301, (2014).

[21] Colbourn, C. J., The Combinatorics of Network Reliability, Oxford University Press, New York, Oxford, (1987).

[22] Cox, D.R. Renewal Theory, In: Methuen's Monographs on

[23] D. E. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, 1989.

[24] Dhillon, B. S., Design Reliability Fundamentals and Applications, 1st ed., CRC Press, Taylor, Francis group, Boca Ration, Florida, (1999).

[25] Dhillon, Balbir S. Maintainability, maintenance, and reliability for engineers. CRC press, (2006).

[26] Diestel, R., Graph Theory. Electronic library of mathematics. Springer Science, (2006).

[27] Dod, M., Domination in graphs with application to network reliability, Hinweis Senior Theses (2015).

[28] Dohmen, K., Inclusion-exclusion and network reliability, The Electronic Journal of Combinatorics, Research Paper R36, 5, 1 – 8, (1998).

[29] Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the 6th international symposium on micro machine and human science, pp 39–43, Nagoya, Japan, Mar 13–16, 1995.

[30] El-Ghazali Talbi," METAHEURISTICS FROM DESIGN TO IMPLEMENTATION", Copyright © by John Wiley & Sons, Inc. All rights reserved 2009.

[31] El-Ghazali Talbi Farouk Yalaoui , “Metaheuristics for Production Systems”, © Springer International Publishing Switzerland 2016.

[32] Faraci, V., Calculating failure rates of series/parallel networks, The Journal of Alion s, System Reliability Center, vol. First Quarter, pp. 1-3, (2006).

[33] Feng, X., Zhu, X., Zhao, W. and Li, X., Reliability of Electric Vehicle with Wind Turbine Based on Particle Swarm Optimization, Chemical Engineering Transactions, 66,1291-1296, (2018),.

[34] Gertsbakh, I. and Shpungin, Y., Network Reliability and Resilience, 1st ed., Springer- Verlag Berlin Heidelberg, (2011).

[35] Gnedenko, B. V. and Ushakov, I. A., Probabilistic Reliability Engineering, John Wiley(1995).

[36] GNEDENKO, B. V. On the limiting distribution of the maximum term in a random series. In: Breakthroughs in statistics. Springer, New York, NY, p. 185-194, 1992..

[37] Gnedenko, B., I. Pavlov, I. Ushakov, (edited by S. Chakravarty), Statistical Reliability Engineering, John Wiley Sons, New York, (1999).

[38] Govil, A. K., Reliability Engineering, TaTa Mc-Graw Hill Pub. Com. Ltd., New Delhi, India, (1983).

[39] Hassan, Z. A. H. and Balan, V., Reliability extrema of a complex circuit on bi-variate slice classes, ScienceDirect. Karbala International Journal of Modern Science, vol. 1.1, pp. 1-8, (2015).

[40] Hassan, Z. A. H. and Mutar, E. K., Evaluation the reliability of a high-pressure oxygen supply system for a spacecraft by using GPD method, Al-Mustansiriyah Journal of college of education, vol. special issue. 2, pp. 993 -1004 , (2017).

[41] Hassan, Z. A. H. and Mutar, E. K., Geometry of reliability models of electrical system used inside spacecraft, 2017 Second Al-Sadiq International Conference on Multidisciplinary in IT and

Communication Science and Applications (AIC-MITCSA). IEEE, pp. 301-306, (2017).

[42] Hassan, Z. A. H. and Udriste, C., Equivalent reliability polynomials modeling EAS and their geometries, Annals of West University of Timisoara Mathematics and Computer Science, vol. 53.1, pp. 177-195, (2015).

[43] Hassan, Z. A. H. and Udriste, C., Geometry of Reliability Models, Ph.D. Thesis, University Politehnica of Bucharest,(2016).

[44] Haynes, T. W., Hedetniemi, S. T. and Slater, P. J., Fundamentals of Domination in Graphs, Marcel Dekker, Inc., New York, (1998).

[45] Hoyland, A. and M. Rausand, System Reliability: Models and Statistical Methods, John Wiley Sons, New York, (1994).

[46] Huang, Chia-Ling, and Wei-Chang Yeh. Simplified Swarm Optimization Algorithm for reliability redundancy allocation problems. International Telecommunication Networks and Applications Conference (ITNAC). IEEE, (2015).

[47] International Electrotechnical Commission. "Functional safety of electrical/electronic/programmable electronic safety related systems." IEC 61508 (2000).

[48] Ireson, W. G., Coombs, C. R. and Moss, R. Y., Handbook of Reliability Engineering and Management, 2nd ed., McGraw-Hill Comp., U.S.A., (1995).

[49] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942–1948, 1995.

[50] J. Kennedy and R. C. Eberhart. Swarm Intelligence. Morgan Kaufmann, San Francisco, CA, 2001.

[51] Jula, Nicolae, and Cepisca Costin. Methods for analyzing the reliability of electrical systems used inside aircrafts. Recent Advances in Aircraft Technology. Intech open, (2012).

[52] K. A. De Jong. Genetic algorithms: A 10 year perspective. In International Conference on Genetic Algorithms, pp. 169–177, 1985.

[53] Katiyar, Sapna. A Comparative Study of Genetic Algorithm and the Particle Swarm Optimization. International Journal of Technology, 2.2: 21-24, (2010).

[54] Kazeminia, A., Reliability Optimization of Hardware Components and System's Topology During Early Design Phase, Ph.D. Thesis, University at Duisburg Essen, (2013).

[55] Kennedy J, Eberhart RC Particle swarm optimization? In: Proceedings of the IEEE international conference on neural networks, Perth, Australia, pp 1942–1948, (1995).

[56] Kim, M. C., Reliability block diagram with general gates and its application to system reliability analysis, Annals of Nuclear Energy, vol. 38.11, pp. 2456-2461, (2011).

[57] Kim, Y. H., Case, K. E. and Ghare, P., A method for computing complex system reliability, IEEE Transactions on Reliability,vol. 21.4, pp. 215-219, (1972).

[58] Ko lowrocki, K., Reliability of large systems and Complex Systems, Wiley Online Library, (2008).

[59] Kulli, V. R., & Soner, N. D. Complementary edge domination in graphs. Indian Journal of Pure and Applied Mathematics, 28, 917-920, (1997).

[60] Kumar, A., On solving complex reliability optimization problem using multi-objective particle swarm optimization. Mathematics Applied to Engineering. Academic Press, 115-131, (2017).

[61] Kumar, A.,.Khosla, A., Saini, J.S. and Singh, S., Meta-heuristic range based node localization algorithm for wireless sensor networks, in localization and GNSS (ICLGNSS), 2012 International Conference on IEEE, pp.1-7, (2012).

[62] Kuo, Way, and Ming J. Zuo. Optimal reliability modeling: principles

and applications. John Wiley, Sons, 2003.

[63] Kuo,W. and Prasad, V. R., An annotated overview of system-reliability optimization, IEEE Transactions on Reliability, 49, 176 – 187, (2000).

[64] Lakey,P. B. and Neufelder, A. M. , System and Software Reliability Assurance Notebook, Rome Laboratory, Rome, NY, 6.1- 6.24, (1996).

[65] Lazzaroni, M., Cristaldi, L., Peretto, L., Rinaldi, P. and Catelani, M., Reliability Engineering Basic Concepts and Applications in ICT, Springer Verlag Berlin Heidelberg, Germany, (2011).

[66] Leemis, L.M, Reliability: Probabilistic Models and Statistical Methods, Prentice-Hall, Englewood Cliffs, NJ, (1995).

[67] Leitch, R.D., Reliability Analysis for Engineers: An Introduction, Oxford University Press, New York, (1995).

[68] Meeker, W. Q. and L. A. Escobar, Statistical Methods for Reliability Data, John Wiley, Sons, New York, (1998).

[69] Marvin Rausand and Arnljot Hoyland. System reliability theory: models, statistical methods, and applications, volume 396. John Wiley & Sons, 2003.

[70] Mettas, A., Reliability allocation and optimization for complex systems, Proceedings Annual Reliability and Maintainability Symposium, Los Angeles, CA, 216-221, January (2000).

[71] Michael O Ball, Charles J Colbourn, and J Scott Provan. Network reliability. Handbooks in operations research and management science, 7:673{762, 1995.

[72] MIL-HDBK-217F. Reliability prediction of electronic equipment, Military handbook. Washington, DC: U.S. Department of Defense. (1995).

[73] MIL-P-1629. Procedures for performing a failure modes, effects, and criticality analysis, Military procedure.Washington, DC: U.S. Department of Defense. (1949).

[74] Muhammad Imran[a] , * , Rathiah Hashima and Noor Elaiza Abd Khalidb," An Overview of Particle Swarm Optimization Variants", Malaysian Technical Universities Conference on Engineering & Technology 2012, MUCET 2012.

[75] Mutar, E. K. and Hassan, Z. A. H., On the Geometry of the Reliability Polynomials, Thesis, University of Babylon, (2017).

[76] Naduvath, Sudev. "Lecture Notes On Graph Theory." (2017).

[77] Nedjah, N., dos Santos, L., Coelho, and de- MacedoMourelle, L., Multiobjective swarm intelligent systems: theory experiences. Springer Science Business Media, vol.261, (2009).

[78] NUREG-75/014. Reactor Safety: An Assessment of Accident Risk in U.S. Commercial Nuclear Power Plants. Report NUREG-75/014.Washington, DC: U.S Nuclear Regulatory Commission. (1975).

[79] Ouyang and Zhiyuan, An improved particle swarm optimization algorithm for reliability redundancy allocation problem with mixed redundancy strategy and heterogeneous components. Reliability Engineering System Safety 181, 62-74, (2019).

[80] Page, L. B. and Perry, J. E., Reliability of directed networks using the factoring theorem, IEEE Transactions on Reliability, vol. 38.5, pp. 556-562, (1989).

[81] PAL, Saibal K.; RAI, C. S.; Singh, Amrit Pal. Comparative study of firefly algorithm and particle swarm optimization for noisy non-linear optimization problems. International Journal of intelligent systems and applications, 4.10: 50, (2012).

[82] Pérez-Rosés, Hebert. "Sixty years of network reliability." Mathematics in Computer Science 12.3: 275-293, (2018).

[83] Rebaiaia, M.L. and AitKadi, D., A new technique for generating minimal cut sets in nontrivial network, AASRI Procedia, vol. 5, pp. 67-76, (2013).

[84] Rosen, K. H., Discrete mathematics and its applications, AMS, (2007).

[85] S. Goss, S. Aron, J. L. Deneubourg, and J. M. Pasteels. Self-organized shortcuts in the Argentine ant. Naturwissenschaften, 76:579–581, 1989.

[86] S.Bitam, M. Batouche, and E.-G. Talbi. A taxonomy of artificial honeybee colony optimization. In E.-G. Talbi and K. Mellouli, editors, META'08 International Conference on Metaheuristics and Nature Inspired Computing, Hammamet, Tunisia, Oct 2008.

[87] S.N. Sivanandam, S. N. Deepa ,“ Introduction to genetic algorithms”,Springer (2008).

[88] Sahoo, Laxminarayan. "Studies on reliability optimization problems by genetic algorithm." (2012).

[89] Sandler, G. H., System Reliability Engineering, Prentice-Hall Int. Series In Space Technology, Prentice Hall Inc., Englewood Cliffs N.J., (1963).

[90] Sanjay Kumar Chaturvedi. Network reliability: measures and evaluation. John Wiley & Sons, 2016.

[91] Shaghaghi, Saba, et al. ”Comparative analysis of GMDH neural network based on genetic algorithm and particle swarm optimization in stable channel design.” Applied Mathematics and Computation 313: 271-286, (2017).

[92] Singh, S., Shivangna, S., Mittal, E., Range Based Wireless Sensor Node Localization Using PSO and BBO and Its Variants. 2013 International Conference on Communication Systems and Network Technologies, 309-315,(2013).

[93] Srinath, L. S., Concepts in Reliability Engineering, East-West Press

Private Ltd., (1985).

[94] Steffen Klamt and Ernst Dieter Gilles. Minimal cut sets in biochemical reaction networks. Bioinformatics, 20(2):226-234, 2004.

[95] Thoft-Cristensen, P. and Baker, M. J., Structural reliability theory and its applications, Springer Science and Business Media, (2012).

[96] Todinov, M. T., Risk-based reliability analysis and generic principles for risk reduction, Elsevier, (2006).

[97] Tummala, Rao R., Eugene J. Rymaszewski, and Y. C. Lee. Microelectronics packaging handbook,241-242, (1989).

[98] Venter, G. and Sobieski, J.,Particle Swarm Optimization, AIAA 2002-1235,43rd IAAA /ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Denver, CO., April (2002).

[99] Voloshin, V. I., Introduction to graph and hypergraph theory, Nova Science Publ., (2009).

[100] Weibull,W. A statistical distribution function of wide applicability. Journal of Applied Mechanics 18: . (1951). 293–297.

[101] Weibull,W. A Statistical Theory of the Strength of Materials. Report 151. Stockholm, Sweden: Royal Swedish Institute for Engineering Research. (1939).

[102] Wilson, Robin J. Introduction to Graph Theory uPDF eBook. Pearson Higher Ed, 2015.

[103] Yamuna, M. and Karthika, K., Medicine Names as a DNA Sequence using Graph Domination, Scholars Research Library Journals, USA, 5(6), 2747-2756, (2014).

[104] Yang, Guang. Life cycle reliability engineering. John Wiley, Sons, (2007).

[105] Yasamin Khorramzadeh. Network Reliability: Theory, Estimation, and Applications. PhD thesis, Virginia Tech, 2015.

[106] Zhang, L., Zhang, N., Zhao, F., and Chen, Y. , A genetic-algorithm-

based experimental technique for determining heat transfer coefficient of exterior wall surface. Applied Thermal Engineering, 24(2-3), 339-349, (2004).

المستخلص

تهدف هذه الأطروحة إلى دراسة النماذج الرياضية في موثوقية الشبكات وكذلك استخدام بعض الأساليب لحساب موثوقية الشبكات المعقدة كتطبيق لنظرية الرسم البياني. سنناقش أيضًا القضايا المهمة التي تشمل: استخدام طريقة رياضية لإيجاد مجموعات المسارات الصغرى والقطوعات الصغرى للشبكة اعتمادًا على جبر المصفوفات، وكذلك استخدام بعض الطرق للحصول على متعددة حدود الموثوقية للشبكة، بما في ذلك: تتبع المسار، الحد الأدنى من القطع طريقة الاختزال إلى عناصر السلاسل وطريقة الشمول والاستبعاد. بالإضافة إلى استخدام بعض التقنيات لحساب تخصيص موثوقية الشبكة المختصرة مثل تقنية التخصيص المتساوي وطريقة ARINC لأسلوب التخصيص وطريقة التخصيص AGREE وتقنية الجدوى للأهداف. تضمنت الأطروحة حساب أهمية جميع مكونات النظام لمعرفة تأثير كل هذه المكونات على عمل النظام ككل.

أخيرًا، سنقوم بحساب تخصيص الموثوقية والتحسين لشبكة معقدة باستخدام الخوارزمية الجينية، وتحسين سرب الجسيمات، وخوارزمية مستعمرة النمل، وتحسين مستعمرة النحل، مع مقارنة بين هذه الخوارزميات لاختيار أفضل خوارزمية تعطي أعلى موثوقية وأدنى كلفة.

جمهورية العراق

وزارة التعليم العالي والبحث العلمي

جامعة بابل

كلية التربية للعلوم الصرفة

قسم الرياضيات

نماذج رياضية لأساليب تحسين الموثوقية للشبكات

أطروحة مقدمة إلى مجلس كلية التربية للعلوم الصرفة في جامعة بابل كجزء من متطلبات نيل درجة الدكتوراه فلسفة في التربية / الرياضيات

من قبل

غازي عبد الله مدلول هدمة

بإشراف

أ.د. زاهر عبد الهادي حسين

1443هـ **2021**م