STUDIES OF
EVOLUTIONARY ALGORITHMS AND APPLICATIONS IN RELIABILITY OPTIMIZATION

Thesis submitted by

AVISHEK BANERJEE

Doctor of Philosophy (Engineering)

Information Technology,
Faculty Council of Engineering & Technology
Jadavpur University
Kolkata, India

2018

Dedicated

to

my parents

Mr. Kalyan Banerjee and Mrs. Sunanda Banerjee

JADAVPUR UNIVERSITY
KOLKATA - 700032, INDIA

INDEX NO. 272/14/E

1. Title of the thesis : **Studies of Evolutionary Algorithms and Applications in Reliability Optimization**

2. Name, Designation & Institution of the Supervisor/s :

   1. Dr. Samiran Chattopadhyay
      Professor, Department of Information Technology
      Jadavpur University

   2. Dr. Asoke Kumar Bhunia
      Professor, Department of Mathematics
      The University of Burdwan

   3. Dr. Laxminarayan Sahoo
      Assistant Professor, Department of Mathematics
      Raniganj Girls' College

3. List of publication:

Published Paper (s)

International Journals

1. L.Sahoo, **A. Banerjee** , A.K. Bhunia and S. Chattopadhyay, An efficient GA-PSO approach for solving mixed integer nonlinear programming problem in reliability optimization, Int. J. **Swarm and Evolutionary Computation, Elsevier**. (2014), Vol. 19, 43–51, December 2014.

2. **A. Banerjee**, S. Chattopadhyay, M. Gavrilas ,and G. Grigorus, Minimization of indices and cost of power distribution systems in urban areas using an efficient hybrid meta-heuristic algorithm. **Soft Computing, Springer** (2017): 1-25.

International conference

1. **A. Banerjee**, M. Gavrilas, O. Ivanov and S. Chattopadhyay, Reliability improvement and the importance of power consumption optimization in wireless sensor networks. 9th International Symposium on Advanced Topics in Electrical Engineering (**ATEE**). **IEEE**, Bucharest , Romania 2015.

2. **A. Banerjee**, M. Gavrilas, G. Grigorus and S. Chattopadhyay, Decision making in assessment of RRAP of WSN using fuzzy-hybrid approach. IEEE International Conference on Advanced Networks and Telecommunications Systems (**ANTS**).**IEEE**, ISI, Kolkata 2015.

3. **A. Banerjee**, S. Chattopadhyay, A. Mukhopadhyay and G. Grigorus, A fuzzy-ACO algorithm to enhance reliability optimization through energy harvesting in WSN, International Conference on Electrical, Electronics, and Optimization Techniques (**ICEEOT**), **IEEE** Chennai, India, 2016, pp. 584-589. doi: 10.1109/ICEEOT.2016.7754748

4. **A. Banerjee**, M. Gavrilas, G. Grigorus and S. Chattopadhyay, A Fuzzy Hybrid Approach for Reliability Optimization Problem in Power Distribution Systems, International conference and Exposition on Electrical and power Engineering , (**EPE**).**IEEE** Iasi , romania 2016.

5. L.Sahoo , **A. Banerjee** , A.K. Bhunia and S. Chattopadhyay , Multi-objective reliability optimization using hybrid algorithm. 1st International Conference on Emerging Trends in Engineering and Science **(ETES)**, 23rd-24th March, 2018, Asansol Engineering College, Asansol, West Bengal.

National conference

1. L.Sahoo, **A. Banerje** , A.K. Bhunia and S. Chattopadhyay, Reliability redundancy allocation problem of series system by Hybrid GA-PSO approach, National Conference on Emerging Trends in Engineering and Science (**ETES**) 2014, Asansol, January 2014.

Communicated Paper (s)

1. L.Sahoo , **A. Banerjee** , A.K. Bhunia and S. Chattopadhyay, Multi-objective reliability optimization problem via Hybrid GA-PSO Algorithm., Int. J. IEEE Transactions on reliability.

4. List of Patents: Nil

5. List of Presentations in National /International/Conferences/Workshops:

International

1. **A. Banerjee**, M. Gavrilas, O. Ivanov and S. Chattopadhyay, Reliability improvement and the importance of power consumption optimization in wireless sensor networks. 9th International Symposium on Advanced Topics in Electrical Engineering (**ATEE**). **IEEE**, Bucharest , Romania 2015.

2. **A. Banerjee**, M. Gavrilas, G. Grigorus and S. Chattopadhyay, Decision making in assessment of RRAP of WSN using fuzzy-hybrid approach. IEEE International Conference on Advanced Networks and Telecommunications Systems (**ANTS**).**IEEE**, ISI, Kolkata 2015.

3. **A. Banerjee**, S. Chattopadhyay, A. Mukhopadhyay and G. Grigorus, A fuzzy-ACO algorithm to enhance reliability optimization through energy harvesting in WSN, International Conference on Electrical, Electronics, and Optimization Techniques (**ICEEOT**), **IEEE** Chennai, India, 2016, pp. 584-589. doi: 10.1109/ICEEOT.2016.7754748

4. **A. Banerjee**, M. Gavrilas, G. Grigorus and S. Chattopadhyay, A Fuzzy Hybrid Approach for Reliability Optimization Problem in Power Distribution Systems, International conference and Exposition on Electrical and power Engineering , (**EPE**).**IEEE** Iasi , romania 2016.

5. L.Sahoo , **A. Banerjee** , A.K. Bhunia and S. Chattopadhyay , Multi-objective reliability optimization using hybrid algorithm. 1st International Conference on Emerging Trends in Engineering and Science **(ETES)**, 23rd-24th March, 2018, Asansol Engineering College, Asansol, West Bengal.

National

1. L.Sahoo, **A. Banerjee**, A.K. Bhunia and S. Chattopadhyay, Reliability redundancy allocation problem of series system by Hybrid GA-PSO approach, National Conference on Emerging Trends in Engineering and Science (**ETES**) 2014, Asansol, January 2014.

ACKNOWLEDGEMENTS

I feel myself not eligible enough unable to adequately thank to my supervisors Dr. Samiran Chattopadhyay, Professor, Department of Information Technology, Jadavpur University and Dr. Asoke Kumar Bhunia, Professor, Department of Mathematics, The University of Burdwan and Dr. Laxminarayan Sahoo, Assistant Professor, Department of Mathematics, Raniganj Girls' College, Raniganj for their valuable guidance, constant help and encouragement throughout my research work. I would like to thank them heartily not only for their scholarly guidance and encouragement, but also for their endless love and support.

I take this opportunity to thank the authorities of the Jadavpur University, Kolkata, India and The University of Burdwan, Burdwan, India and Asansol Engineering College, Asansol, India and Raniganj Girls' College, Raniganj, India and 'Gheorghe Asachi' Technical University of Iasi, Romania for providing me the opportunity to carry out this research work and for extending whole hearted cooperation and support.

I sincerely acknowledge all the help, cooperation and constructive suggestions from the Head of the Department and all other faculty members and staff of the Department of Information Technology, Jadavpur University, during the course of my research work. I would also like to extend my thanks especially to Prof. univ. Dr. ing. Mihai Gavrilas, Professor, Department of Power Engineering, Technical University of Iasi, Romania and for his immense cooperation, guidance and valuable suggestions during my visit in Romania to obtain the Erasmus Mundus Scholarship in TUIASI, Romania.

I feel privileged to thank Dr. Debashis Sarkar, Principal, Asansol Engineering College, for his cooperation and valuable suggestions.

I would like to express my heart-felt thanks to all of my colleagues and staff of Asansol Engineering College for their encouragement, support and cooperation. In particular, I thank Asist. Dr. ing. Ovidiu Ivanov, Asst. Professor, Department of Power Engineering, Technical University of Iasi, Romania and Prof. univ. Dr. ing. Gheroghe Grogoras, Professor, Department of Power Engineering, Technical University of Iasi, Romania for transferring knowledge about various engineering topics, which laid to write different articles in various renowned conference and journal.

My sincere thanks also go to HOD In-charge, the department of Information Technology, Asansol Engineering College, Mr. Anup Kumar Mukhopadhyay, for his very sincere encouragement and helpful suggestions.

I would like to express my gratitude to Dr. Pintu Pal, Assistant Professor, Asansol Engineering College, Dr. Sanat Kr. Mahato, Associate Professor, Sidho-Kanho-Birsha University and Dr. Akbar Ali sheikh, Assistant Professor, The University of Burdwan for their constant help and suggestions.

I also acknowledge the financial support by EU Erasmus Mundus Project – cLINK(Center of Excellence for Learning, Innovation, Networking and Knowledge), Grant Agreement No: 212-2645/001-001-EM, Action-2 Partnerships. The project is developed under “Erasmus Mundus External Corporation: Asia Region, Lot 9 (Ref. EACEA/42/11)” and it is coordinated by the university of Northumbria from Newcastle, UK. The partner university is TUIASI, Romania, and the research work was performed in its Power Engineering department of the Electrical Engineering Faculty. I am thankful to EU Erasmus Mundus Project for providing financial scholarship support and got the opportunity to visit ‘Gheorghe Asachi’ Technical University of Iasi, Romania for period of 23 months (August 2014 to June 2016).

My deepest appreciation goes to my parents, my wife Mrs. Sumana Banerjee, my little son Soumava and little nephew Ritobrata. Last but not the least my heartiest thanks go to my sister Mrs. Antara Mukherjee, brother-in-laws Mr. Subhajit Chakraborty, Mr. Ritwik Mukherjee and my in laws Mr. Jadu Nath Chakraborty and Mrs. Shrabani Chakraborty. I could not have finished this research work without their unending love.

In fine, I am solely responsible for any errors and omissions in this thesis.

Place: Jadavpur University **(Avishek Banerjee)**

Date: 16/07/2018

CONTENTS

Page No.

Page No.

NOTATIONS USED IN THE THESIS

x	$(x_1, x_2, \ldots, x_n)$, is the vector of the redundancy allocation of the system
r	$(r_1, r_2, \ldots, r_n)$, the vector of the component reliabilities for the system
n	the number of subsystems in the system
(x, r)	The vector of the redundancy allocation and component reliabilities of the system
x_j	The number of redundancy of the sub system $j\,(j = 1, 2, \ldots, n)$
R_s	The system reliability
g_i	The $i-th$ constraint function
b_i	The upper limit on the $i-th$ resource
w_i	The component weight of $i-th$ subsystem
v_i	The component volume of $i-th$ subsystem
c_i	The component cost of $i-th$ subsystem
W	The upper bound on the weight of the system
V	The upper bound on the volume of the system
C	The upper bound on the cost of the system
Z^+	Set of non-negative integers
P_s	Population size
M_g	Maximum number of generations
P_m	Probability of mutation
P_c	Probability of crossover
$U(\eta, \xi)$	Uniform distribution between η and ξ
PSO-Co	Constriction coefficient-based Particle Swarm Optimization
$\tilde{r}$	Fuzzy valued vector of the component reliabilities for the system
$\tilde{R}_j$	The fuzzy valued overall system reliability
$\tilde{w}_j$	The fuzzy valued component weight of subsystem j
$\tilde{v}_j$	The fuzzy valued component volume of subsystem j
$\tilde{W}$	The fuzzy valued upper bound on the weight of the system
$\tilde{V}$	The fuzzy valued upper bound on the volume of the system
$\tilde{C}$	The fuzzy valued upper bound on the cost of the system
α_j, β_j	The physical features of system components
W'	Left tail statistic in Wilcoxon Rank-Sum test
W"	Right tail statistic in Wilcoxon Rank-Sum test
W-crit	Tail statistic with critical value in Wilcoxon Rank-Sum test
α	Tail value(constant) for tail test in Wilcoxon Rank-Sum test
p-value	Probability value in Wilcoxon Rank-Sum test

TABLES

FIGURES

ACRONYMS USED IN THE THESIS

CPU time	CPU Execution time to execute some computer program
GA	Genetic Algorithm
PSO	Particle Swarm optimization
DE	Differential Evolution
ACO	Ant Colony Optimization
INLPP	Integer Non-linear Programming Problem
MINLPP	Mixed-integer Non-linear Programming Problem
TFN	Triangular Fuzzy Number
PFN	Parabolic Fuzzy Number
TrFN	Trapezoidal Fuzzy Number
max_gen	Maximum Number of Generation
QPSO	Quantum behaved Particle Swarm optimization
WQPSO	Weighted Quantum behaved Particle Swarm optimization
AQPSO	Adaptive Quantum behaved Particle Swarm optimization
P_C	Crossover Probability
P_m	Mutation Probability
P_s	Population Size
PC	Personal Computer
RAP	Redundancy Allocation Problem
WSN	Wireless Sensor Network
PDS	Power Distribution System

CHAPTER 1

Introduction

1.1 General Introduction

Evolutionary Computation is a type of computing which evolves its basic characteristics using principles inspired from biological evolution processes. The basic idea of the evolutionary computing relates with the powerful natural evolution processes to solve real life optimization problems [1] by trial and error method.

Evolutionary computational [2] models are referred as evolutionary algorithms (EAs) that can be easily expressed as follows:

Generate a population of samples (i.e., chromosomes or particles) carrying the characteristics of the formulated problem under their structure (i.e., in genes)

Repeat the following steps until the stopping condition(s) is/are satisfied

i) Test the structure of the sample(s) for quality

ii) Select samples to reproduce new variations of the selected structures

iii) Replace old sample with a new one, of better structure

EAs refer to a generic meta-heuristic optimization algorithms characterised by implementations looking at a guided random search of an iterative processes. EAs include a family of heuristic algorithms called meta-heuristics. Nowadays EAs has become a more powerful research area in computer science as well as different branches of science and technologies.

EAs started with three research topic in the 1950s and 1960s: genetic algorithms, evolution strategy and evolutionary programming.

Hybrid Algorithms [3] combine two or more algorithms to solve optimization problems which are more complicated and cannot be solved easily by using any other direct/indirect optimization techniques. Generally the main goal is to combine the entire features/properties of each algorithm to get higher performance in different aspects viz. time complexity, space complexity as well as to reduce the computational cost. The combined strategies may be applied sequentially or switching between them over the execution of the algorithm.

The reliability of a system[4] can be measured by the probability of success in given operating conditions, over a specified time period, generally termed as “mission time”, whereas Reliability Optimization imposes some merit functions to be achieved, like maximization of system reliability and the minimization of system cost, minimization of system volume and minimization of system weight. The reliability constraints are typically modelled as probability of failure / repair or other environment variables.

The main objective of reliability optimization [5] is to increase the system reliability. This can be done in different ways which are as follows:

(i) Increasing the component/sub-system reliability of the system/network.

(ii) Using repair or maintenance mechanisms where failed/damaged components are replaced/repaired.

(iii) Using standby redundancy which is switched to active when a failure occurs.

(vi) Using better arrangement for exchangeable/alternative components.

(v) Using preventive maintenance such that components are replaced by new ones whenever they fail or at some fixed interval, whichever is earlier.

To implement the above steps, the construction of the resource reserve is an important step and there should be a balance between the use/consumption of resources and the improvement of system reliability.

When, redundancy is used to improve the system reliability, the corresponding problem is known as the redundancy allocation problem. The objective of this problem is to find out the number of redundant components that maximizes the system reliability and the minimization of system cost, volume and weight under several resource constraints. This problem is studied since 1950s, because of its potential broad applications. When it is difficult to improve the reliability of unreliable components, system reliability can easily be enhanced by adding redundancies on those components. However, for design engineers, improving of component reliability has been generally preferred over adding redundancy, because, in many cases, redundancy is difficult to add with real systems due to technical limitations and relatively large amount of required resources, such as weight, volume and cost.

There are many applications in different engineering fields such as telecommunications, computer networking, electrical networks, gas sewer networks, etc. where reliability optimization can be more appropriate for designing the system.

To efficiently build fault-tolerant systems [6] with redundancy, the number of redundancies should be optimized. However, for improving the system reliability, the addition of redundant components to the system is a difficult task; due to several constraints arising out of the size, cost and quantities of resources coupled with technical constraints. Thus, the redundancy allocation problem can be seen as a practical problem of determining the appropriate number of redundant components that maximize the system reliability under different resource constraints. Irrespective of those physical scalar requirements like size, cost, weight and number of resources, here we have been able to consider some characteristic requirements in our research. So, in most of the cases, the problem has been formulated as a non-linear constrained optimization problem [7] with integer/mixed-integer variables. Also, some researches have been done under fuzzy environment. To solve this type of problem, several researchers have proposed different approaches. Generally, in their works, the parameters of the system are assumed to be known precise valued. However, in real-life circumstances, the design parameters may fluctuate due to some environmental conditions. Hence, it is sensible to consider the design parameters as imprecise numbers. To define the problem associated with such imprecise numbers, different approaches like stochastic, fuzzy and fuzzy-stochastic approaches are used. Here, we have used fuzzy approach to handle the impreciseness. As a result, the objective function as well as constraints of the formulated problem will be fuzzy valued, and these problems need to be optimized.

These types of optimization problems with precise valued/fuzzy valued objective functions along with constraints can be solved by a well-known powerful computerized heuristic search and optimization methods, i.e., Evolutionary Algorithms (EA) like Genetic Algorithm (GA), Particle Swarm

Optimization (PSO), Differential Evolution (DE), Ant Colony Optimization (ACO) etc.

This thesis is mainly focused on two different and important engineering fields, i.e., Wireless Sensor Networks (Modern Communication Technology) and Power Distribution Systems (Power Engineering). Recent advanced technologies have rejuvenated the importance of the redundancy strategy on these application fields. The current economizing trend in the system design and manufacturing has caused many unavoidable faults/defects which have been approached in the thesis using the concept of fuzziness which are studied in different researches [5]. It is widely known that there are certain limitations on enhancing reliability or yield in system design/manufacturing by developing relevant technologies. Hence, various fault-tolerant and self-repairable techniques have been well studied. These approaches are mainly based on adding redundancies on components and controlling the usage of redundancies.

1.2 Basic Definitions, Theory and Terminologies

1.2.1 Reliability Definition

According to the Rausand [8], the definition of reliability is as follows:

"Reliability is the probability that a system will perform satisfactorily for at least a given period of time when used under stated conditions".

So, the reliability is defined as the probability of a device performing its intended purpose adequately for the period of time under the operating conditions encountered. The reliability is the probability with which the devices will not fail to perform a required operation for certain duration of time. This definition brings into the focus four important factors which are as follows:

(i) The reliability of a device is expressed as a probability.

(ii) The device is required to give adequate performance.

(iii) The duration of adequate performance is specified.

(iv) The environmental or operating conditions are specified.

However, in practice, even the best design manufacturing and maintenance efforts do not completely eliminate the occurrence of failure.

1.2.2 System Reliability

According to Kuo [9], “System reliability is a measure of how well a system meets its designed objective and it is usually expressed in terms of the reliabilities of the subsystems of components”.

Generally, to determine the reliability factor the system is splited into subsystems and elements whose individual reliability factors can be estimated or determined. Depending on the manner in which these subsystems and elements are connected to constitute the given system, combinatorial rules are applied to obtain the system reliability.

1.2.3 Fundamental System Configurations

In many cases, a system is not constructed using a single component. We always want to evaluate the reliability of a simple as well as complex/complicated system. Let us consider a reliability system consisting of a number of components. These components may be hardware, human or even software also. If some of the components are software products, then their modelling requires special attention.

Here, we shall discuss some important reliability configurations which are as follows:

1.2.4 The Series Configuration

The series configuration is the simplest and perhaps one of the most common structures. In this configuration, all the components must operate in order to ensure the system operation. In other words, the system fails when any one of the components fails.

1.2.5 The Parallel Configuration

A parallel configuration is a system that is not considered to have failed unless all components have failed. This is sometimes called a redundant configuration. The word “redundant” is used only when the system configuration is deliberately changed to produce additional parallel paths in order to improve the system reliability. In a parallel configuration consisting of a number of components, the system works if any one of those components is working.

1.2.6 The Series-Parallel Configuration

Let us consider a system which consists of k subsystems connected in parallel, with the *i-th* subsystem consisting of n_i series components, for $i = 1, 2, \cdots, k$. Such a system is called a series-parallel system [10].

1.2.7 The Parallel-Series Configuration

Let us consider a system consisting of k subsystems in series and subsystem *i,* $1 \le i \le k$, has n_i components in parallel. Such a system is called a parallel-series system [11].

1.2.8 Hierarchical Series-Parallel Systems

A system is called a hierarchical series-parallel system (HSP) [12] if it can be viewed as a set of subsystems arranged in a series-parallel pattern; each subsystem has a similar configuration; subsystems of each subsystem have a similar configuration and so on. This system has a non-linear and non-separable structure and consists of nested parallel and series systems.

1.2.9 The Complex/Complicated/Bridge System

Sometimes a system cannot be reduced to series and parallel configurations, because it contains combinations of components which are connected neither in a series nor in parallel; those systems are called complex/complicated/bridge or non-parallel series systems.

1.2.10 The K-out-of-N System

A $k-out-of-n$ system [13] is an n-component system which functions when at least k components out of n function satisfactorily. This redundant system is sometimes used in the place of a pure parallel system. It is also referred to as $k-out-of-n:G$ system. An n-component series system is a $n-out-of-n:G$ system whereas a parallel system with n-components is a $1-out-of-n:G$ system.

1.3 The Objectives and the Motivation of the Thesis

The primary objectives of thesis are as follows:

1.3.1 Development of hybrid algorithms based on GA with advanced operators

Sometimes genetic algorithm does not reach to the peak. In such cases, all or most of the chromosomes of the populations concentrate on a small part of the search space located around the local optima. This is known as premature convergence. To avoid this premature convergence, genetic algorithm is combined with another algorithm to increase the efficiency, accuracy and consistency of the developed algorithm. This is known as hybrid algorithm. Our aim is to develop new hybrid algorithm based own GA with advanced/newly proposed operators for solving optimization problems in crisp and/or fuzzy environments.

1.3.2 Development of particle swarm optimization (PSO) based on quantum mechanics

Recently a new version of PSO, called quantum-behaved particle swarm optimization (QPSO) has been proposed in order to improve the global search performance of the original PSO. Global convergence is guaranteed with QPSO, whereas this is not the case with the original PSO. In QPSO, particles' state equations are structured by wave function and each particle state is described by the attracter and the characteristic length of δ-trap. Moreover, the QPSO has fewer parameters to control which makes it easier to implement. Our goal is to develop advanced/improved QPSO to increase the efficiency of the algorithm in solving optimization problems, in crisp and interval environment.

1.3.3 Development of improved Ant Colony Optimization and Differential Evolution

Like genetic algorithm, ant colony optimization and differential evolution belong to the same class of algorithms. However, the upgrading strategies of these algorithms in each iteration are different. In this thesis, our aim is to develop some hybrid algorithms based on either ant colony optimization or differential evolution or both for solving optimization problems.

1.3.4 Application of above mentioned algorithms/methods in reliability optimization.

In all the existing works (except few) in the area of reliability optimization and related problems, the reliabilities of components of a system are assumed to be precise positive numbers which lie between zero and one. However, in reality, the reliability of an individual component may not be precise. It may fluctuate due to several reasons. So the reliability of each component is sensible and it may be treated as a positive imprecise number. To tackle the problem with such imprecise numbers, several approaches like fuzzy, stochastic, fuzzy stochastic and interval approaches are applied. In this work, our aim is to solve the reliability optimization problems and related problems with précised valued/fuzzy valued parameters by different methods mentioned in (1.3.1) – (1.3.3). Here it is mentioned that, we have also formulated and solved two engineering application problems viz. Application of reliability redundancy allocation problem in Wireless Sensor Network (WSN) and reliability optimization in Power distribution System (PDS) by using different methods mentioned in (1.3.1) – (1.3.3).

1.4 Organization of the Thesis

In this thesis we have developed two hybrid algorithms namely GA-PSO and GA-ACO and we have formulated and solved some reliability optimization problems. Also we have applied these algorithms in wireless sensor network problems and problems related with power distribution system in electrical network system. The thesis has been divided into eight chapters as follows:

Chapter 1 : Introduction

Chapter 2 : Literature Review

Chapter 3 : Solution Methodology

Chapter 4 : GA-PSO Algorithm for mixed-integer nonlinear programming problem in reliability optimization

Chapter 5 : Multi-objective reliability optimization problem via Hybrid GA-PSO Algorithm

Chapter 6 : The Reliability Redundancy Allocation Problem and its Application in Wireless Sensor Networks (WSN)

Chapter 7 : Reliability Optimization in Power Distribution Systems (PDS) using hybrid GA-ACO algorithm

Chapter 8 : General Conclusion and Scope of Future Research

Chapter 2 makes an overview of past and recent developments on different evolutionary algorithms. In this chapter, the discussion is mainly focused on a literature survey about Evolutionary Algorithms (EA), Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), Ant Colony Optimization (ACO), Reliability Optimization Problem (ROP), Wireless Sensor Network (WSN) and Power Distribution System (PDS).

Chapter 3 deals with an overview of existing finite interval mathematics, and fuzzy sets, defuzzification processes. In this chapter, we have also discussed about genetic algorithms (GA), particle swarm optimization (PSO), Differential Evolution (DE) , Ant colony optimization (ACO) , GA-PSO hybrid algorithm and GA-ACO hybrid algorithm

The objective of **Chapter 4** deals with the development of an efficient hybrid approach based on genetic algorithm and particle swarm optimization for solving mixed integer nonlinear reliability optimization problems in series, series-parallel and bridge systems. This approach maximizes the overall system reliability subject to the nonlinear resource constraints arising on system cost, volume and weight. To meet these purposes, a novel hybrid algorithm with the features of advanced genetic algorithm and particle swarm optimization has been developed for determining the best found solutions. To test the capability and effectiveness of the proposed algorithm, three numerical examples have been solved and the computational results have been compared with the existing ones. From comparison, it is observed that the values of the system reliability are better than the existing results in all three examples. Moreover, the values of average computational time and standard deviation are better than the same of similar studies available in the existing literature. The proposed approach would be very helpful for reliability engineers/practitioners for better understanding about the system reliability and also to reach a better configuration.

Chapter 5 presents multi-objective reliability-redundancy allocation problem by hybrid optimization techniques. This technique is based on combination of genetic algorithm and particle swarm optimization. In this chapter, we have solved mixed-integer nonlinear multi-objective reliability optimization problem. The Reliability optimization problem involves selection of components with different choice in redundant levels as well as reliability level to optimize the design problems with respect to several constraints viz. cost, weight and volume. Basically, the main task of the problem is to maximize the overall system reliability and minimize the system cost, system weight and volume of the system. In this chapter we have solved such types of optimization problem and for this purpose we have formulated multi-objective optimization problem considering objective functions as minimize the system cost, minimize the system volume and minimize the system weight respectively along with the restriction on targeted system reliability which is the only constraint of the problem. Here we have solved the problem using hybrid GA-PSO algorithm. Finally, to find out the optimum result and to test the effectiveness of the GA-PSO algorithm, numerical example has been solved and computed results have been presented.

Chapter 6 focuses on Reliability Redundancy Allocation Problem (RRAP) in Wireless Sensor Network (WSN) system is obviously an important problem. The basic function of WSN system is to provide surveillance data transmission over a specified area maintaining minimum power consumption (minimum cost), occupying minimum volume and weight of system components with a reasonable level of reliability. In this chapter, a decision making assessment of reliability of Redundancy Allocation Problem (RAP) is proposed using fuzzy approach. The fuzzy approach incurs the virtue of uncertainty in account to make the approach more practical. Triangular Fuzzy membership function is introduced to produce fuzzy number set as input variables (cost, weight and volume) to a hybrid optimization algorithm. A hybrid meta-heuristic algorithm aiming for reliability optimization in RAP of system components of WSN is discussed. This algorithm is based on a new hybrid algorithm using Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). The fuzzy results obtained are used to exhibit decision making matrix to enhance decidability property of WSN. Finally after defuzzification crisp data are obtained and compared with other approaches from literature and found satisfactory.

In Chapter 7, Power Distribution Systems (PDS) in urban areas suffer from different types of problems. One such major problem is accidental or scheduled interruption. In electrical networks, effects of interruptions are usually quantified using a set of reliability indices, namely, the System Average Interruption Frequency Index (SAIFI) and the System Average Interruption Duration Index (SAIDI). Installation cost (fixed cost) and cost due to temporary and / or permanent faults during interruptions (variable cost) are also major issues to be considered while achieving a cost efficient, fault tolerant PDS. Formalization of an optimization problem that jointly minimizes the afore-mentioned reliability indices as well as the cost of a PDS by optimal allocation of different protective devices and switches has always been a challenging task. This chapter presents a hybrid single as well as joint-objective function optimization technique to minimize different reliability indices (mixed-integer minimization problems), as well as the operational cost of a PDS in urban areas. In the proposed technique, two well-known meta-heuristic search techniques, namely Genetic Algorithms (GA) and Ant Colony Optimization (ACO), have been hybridized after modifying different participating operators. The effectiveness of the proposed algorithm is examined and each PDS is tested in a different environment of constrained optimization. In addition, the presented simulation results are compared with existing approaches that solve this problem. The simulation results show the superiority of the proposed hybrid GA-ACO model, as compared to other established heuristic approaches.

In **Chapter 8,** general concluding remarks drawn from this research studies and the future scope of research have been furnished.

CHAPTER 2

Literature Survey

In this chapter, we have briefly discussed about literature survey on different Evolutionary Algorithms and reliability optimization problems. Here, we have also discussed about literature survey of Wireless Sensor Network (WSN) and Power Distribution System (PDS)

2.1 Literature Survey on Evolutionary Algorithms (EA)

Since late 1950's, Evolutionary Algorithms were pointed out by so many influencing works of Bremermann [14], Friedberg[15,16], Box [17], and others, but the field remained relatively unknown to the broader scientific community for almost few decades. The lack of available powerful computer platforms at that time and methodological shortcomings of those early approaches may be some reasons for implementing these algorithms (see, e.g., Fogel [18]). The fundamental work of Holland [19], Rechenberg [20], Schwefel [21], andFogel [22] served to slowly change this picture during the 1970's, and currently we have observe a remarkable and steady (still exponential) increase in the number of works (see, e.g., Alander [23]) in this field and a clear demonstration of the scientific as well as economic relevance of this subject matter.

The majority of current implementations of evolutionary algorithms descend from three strongly related, but independently developed approaches: viz., Genetic Algorithms, Evolutionary Programming, and Evolutionary Strategies. Genetic Algorithm was first introduced by Holland [19, 24, 25], and effectively studied by De Jong [26-29], Goldberg [30-34], Davis [35], Eshelman [36, 37], Forrest [38], Grefenstette [39-42], Koza [43, 44], Mitchell[45], Riolo [46,47], and Schaffer [48-50]. EAs have been originally proposed as a general model of adaptive process, but by far the largest application of these techniques is in the optimization domain [28, 29]. Since this is true for all three of the mainstream algorithms presented in this thesis, we have discussed their capabilities and performance mainly as optimization strategies.

Evolutionary programming, introduced by Fogel [22, 51] and extended in Burgin [52, 53], Atmar [54], Fogel[55-57], and others, was originally offered as an attempt to create artificial intelligence.

Evolutionary strategies, as developed by Rechenberg [58, 59] and Schwefel [60, 61], and extended by Herdy [62], Kursawe [63], Ostermeier [64,65], Rudolph [66], Schwefel [67], and others, were initially designed with the goal of solving difficult discrete and continuous, mainly experimental [68], parameter optimization problems. During the 1980's, advances in computing system enabled the application of Evolutionary Algorithms to solve difficult real-world optimization problems and solutions have been received for a broader scientific community. In addition, beginning in 1985, international conferences on the techniques were established (mainly focusing on Genetic Algorithms [69-74], with an early emphasis on Evolutionary Programming [75-79], as small workshops on theoretical aspects of Genetic Algorithms [80-82], as a Genetic Programming conference [83], with the general theme of nature-inspired problem solving methods [84-87], and with the general topic of Evolutionary Computation [88-91]). But, somewhat surprisingly, the researchers in the various disciplines of evolutionary computation remained isolated from each other until the meetings in the early 1990's [72, 76, 84]. The remainder of this section is intended as an overview of the current state of the field. We cannot claim that this overview is close to complete. As good starting points for further studies, we referred to the works of [18, 31, 35, 45, 46, 61], and [92-95].

In 1998, Fogel highlights in his book, "Evolutionary Computation: The Fossil Record" [96], how some early ideas have developed into the current thinking and how others have been lost and await rediscovery. The introductions to each chapter reflect Fogel's one-on-one conversations with the authors and their colleagues, conducted over a period of four years. Evolutionary Computation: The Fossil Record provides in-depth historical information and technical detail that is simply unmatched in the field. This book is complete with an extensive bibliography of related literature. Evolutionary Computation: The

Fossil Record may be of particular interest to researchers and students in need of a comprehensive resource on this fascinating area of computer science.

In the year 1999, Jürgen Branke surveys [97] a number of approaches that extend the Evolutionary Algorithm with implicit or explicit memory and suggests a new benchmark problem. In the year 2000, Bäck, Fogel and Michalewicz wrote a book named "Evolutionary Computation: Basic Algorithms and Operators" [98] that enlightened the vast area of evolutionary computation. In the year 2003, N. Sinha , R Chakraborty and P. K. Chattopadhyay emphasized in a well written paper named "Evolutionary Programming Techniques for Economic Load Dispatch " [99], and in this paper they proposed various modifications to the basic method of Evolutionary Algorithms to enhance speed and robustness, and these methods have been applied successfully on some benchmark mathematical problems, as well as few applications on real-world problems.

One main difficulty in applying EAs to real-world problems is that they usually need a large number of fitness evaluations before a satisfying result can be obtained. In the year 2005, Yaochu Jin published in one of his papers [100] "A Comprehensive Survey of Fitness Approximation in Evolutionary Computation", research about the Fitness Approximation concept.

In the year of 2006, KA De Jong wrote a book entitled "Evolutionary Computation: A Unified Approach" [101] describing the canonical as well as the unified view of evolutionary algorithms.

In the year of 2010, Dudy Lim, Yaochu Jin published a paper "Generalizing Surrogate-Assisted Evolutionary Computation" [102], describing a generalization of surrogate-assisted evolutionary frameworks for optimization of problems with objectives and constraints that are computationally expensive to evaluate.

In the year of 2011, Yaochu Jin published a survey paper entitled "Surrogate-Assisted Evolutionary Computation: Recent Advances and Future Challenges" [103] emphasizing the idea of evolutionary computation in a different approach. In the year of 2012, T Bäck and HP Schwefel discussed in one of their papers, "An Overview of Evolutionary Algorithms for Parameter Optimization", about a comparative study of evolutionary computation methods [104].

In the year of 2013, Dasgupta Dipankar and Zbigniew Michalewicz wrote a book titled "Evolutionary Algorithms in Engineering Applications" [105] to solve various engineering problems with the help of Evolutionary Algorithms.

2.2 Literature Survey on Genetic Algorithms (GA)

Genetic Algorithm (GA) is one of the most efficient and powerful heuristic search optimization methods based on the mechanics of natural genetics and natural selection which imitate the Darwin's evolutionary principle "Survival of the fittest". Prof. J. H. Holland [106] first developed the concept of genetic algorithm. Thereafter, a large number of works have been done for the development of genetic algorithm. GA provides good solutions to many complicated optimization problems and received significant attentions during the last four decades. When the objective functions in the optimization problems are multi modal or the search space are irregular, GA need to be highly robust in order to avoid finding stuck at a local optimal solution. The main advantage of GA is just able to find out the global optimal solution. Furthermore, GA does not require the particular mathematical analysis of the optimization problems , which makes GA easily coded by users who are not necessarily good knowledge of mathematics.GA has been well studied in the literature, such as in Holland [106] , Schaffer [107], Michalewicz [108], Koza [109], Liu[110] and have been applied to a wide range of problems.

Very recently GA has been successfully applied in literature in different fields such as networking problem, inventory control theory, game theory, scheduling problem, graph theory, reliability optimization, wireless sensor network, power distribution system in electrical network etc. For details about these one may refer to the works of Bielli and Carotenuto [111], Holland [112], Anderson [113], Kadri and Boctor [114], Corus and Lehre [115], Kim and Kim [116], Biswas et al. [117], Abdul et al. [118].

2.3 Literature Survey on Particle Swarm Optimization (PSO)

The term 'particle' means any natural agent that describes the `swarm' behaviour. The PSO model is a particle simulation concept, and was first proposed by Eberhart and Kennedy [119]. Based upon a mathematical description of the social behaviour of swarms, it has been shown that this algorithm can efficiently find good solutions to a certain number of complicated situations such as, for instance, static optimization problems, topological optimization and others (Parsopoulos, K.E. et al.) [120]; (Parsopoulos, K.E.et al.) [121]; (Fourie, P.C. et al.,) [122]; (Fourie, P.C. et al.,) [123]. Since then, several PSO variants have been developed (Eberhart,R.C. et al) [124]; (Kennedy, J. et al.) [125]; (Kennedy, J. et al.,) [126]; (Shi, Y. et al.) [127]; (Shi, Y.H. et al.) [128]; (Clerc, M.) [129]. It has been shown that convergence of the PSO algorithm is implicitly guaranteed if its parameters are adequately selected (Eberhart, R.C. et al. [130]). Several kinds of problems solving start with computer simulations in order to find and analyze the solutions which do not exist analytically or have been proven to be theoretically intractable. This is easy to understand by the work of Engelbrecht[131].

2.4 Literature survey on Differential Evolution (DE) and related developments

The DE [132-136] algorithm emerged as a very competitive form of evolutionary computing more than a decade ago. The first written article on DE appeared as a technical report by Storn and Price [134] in 1995. One year later, the success of DE was demonstrated in 1996, at the First International Contest on Evolutionary Optimization, which was as held in conjunction with the 1996 IEEE International conference on Evolutionary Computation (CEC) [135]. DE finished third at the First International Contest on Evolutionary Optimization (1st ICEO), which was held in Nagoya, Japan. DE turned out to be the best evolutionary algorithm for solving the real-valued test function suite of the 1st ICEO (the first two places were given to non-evolutionary algorithms, which are not universally applicable, but solved the test-problems faster than DE). Price presented DE at the Second International Contest on Evolutionary Optimization in 1997 [132] and it turned out as one of the best among the competing algorithms. Two journal articles [133, 137] describing the algorithm in sufficient details followed immediately in quick succession. In the year of 2005 on the CEC competition on real parameter optimization on 10-D problems, the classical DE secured the 2nd rank and a self-adaptive DE variant called SaDE [138] secured the third rank, although they performed poorly over 30-D problems. Although a powerful variant of ES, known as Restart Covariance Matrix Adaptation ES (CMA-ES) [139,140] yielded better results than the classical and self-adaptive DE, later on, many improved DE variants like improved SaDE [141], jDE [142], opposition-based DE (ODE) [143], DE with global and local neighborhoods (DEGL) [144], JADE [145], and so on, that will be discussed in subsequent sections] were proposed in the 2006–2009 period. Hence, another rigorous comparison is needed to determine how well these variants might compete against the restart CMA-ES and many other real parameter optimizers over the standard numerical benchmarks. It is also interesting to note that the DE variants continued to secure front ranks in the subsequent CEC competitions [146], like the CEC 2006 competition on constrained real parameter optimization (first rank), the CEC 2007 competition on multi-objective optimization (second rank), the CEC 2008 competition on large scale global optimization (third rank), the CEC 2009 competition on multi-objective optimization (first rank was taken by a DE-based algorithm MOEA/D for unconstrained problems), and in the CEC 2009 competition on evolutionary computation in dynamic and uncertain environments (first rank). We can also observe that no other single search paradigm such as PSO was able to secure competitive rankings in all CEC competitions. A detailed discussion on these DE-variants for optimization in complex environments will be provided in chapter 3. In the DE community, the individual trial solutions (which constitute a population) are called parameter vectors or genomes. DE operates through the same computational steps as employed by a standard EA. However, unlike traditional EAs, DE employs difference of the parameter vectors to explore the objective function landscape. In this respect, it owes a lot to its two ancestors namely—the Nelder-Mead algorithm [147], and the Controlled Random Search (CRS) algorithm [148], which also relied heavily on the difference vectors to perturb the current trial solutions. Since late 1990s, DE started to find several significant applications to optimization problems arising from diverse science and engineering domains. Below, we point out some of the reasons why researchers have been looking at DE as an attractive optimization tool and, as we shall proceed through this survey, these reasons will become more obvious.

Recently, Differential Evolution with dynamic parameters selection for optimization problems has been studied by Sarker et al. [149].

2.5 Literature survey and related developments on Ant Colony optimization (ACO)

Ant Colony Optimization (ACO) [150-152] is a meta-heuristic for solving hard combinatorial optimization problems. The inspiring source of ACO is the pheromone trail laying and following behaviour of real ants, which use pheromones as a communication medium. In analogy to the biological example, ACO is based on indirect communication within a colony of simple agents, called (artificial) ants, mediated by (artificial) pheromone trails. The pheromone trails in ACO serve as distributed, numerical information, used by the ants to probabilistically construct solutions to the problem being solved and in which the ants adapt during the algorithm's execution to reflect their search experience. The first example of such algorithm is the Ant System (AS) [153-156], which was proposed using an example of the well-known Travelling Salesman Problem (TSP) [157]. Despite encouraging initial results, AS could not compete with state-of-the-art algorithms for the TSP. Nevertheless, it had the important role of stimulating further research both on algorithmic variants, which obtain much better computational performance, and on applications to a large variety of different problems. In fact, now exist a considerable number of applications of such algorithms where world class performance is obtained. Examples are applications of ACO algorithms to problems such as sequential ordering [158], scheduling [159], assembly line balancing [160], probabilistic TSP [161], 2D-HP protein folding [162], DNA sequencing [163], protein–ligand docking [164], packet-switched routing in Internet-like networks [165], and so on. The ACO meta-heuristic provides a common framework for the existing applications and algorithmic variants [150,151]. Ant colony optimization for mixed-variable optimization problems has been proposed by Liao et al. [166]. A survey: Ant Colony Optimization based recent research and implementation on several engineering domain has been proposed by Mohan et al. [167].

2.6 Literature survey on reliability optimization problems

According to the types of decision variables, Reliability Optimization problems can be classified into three categories. These are reliability allocation, redundancy allocation and reliability redundancy allocation problems. Classification of reliability optimization problems according to the decision variables has been shown in Figure 2.1.

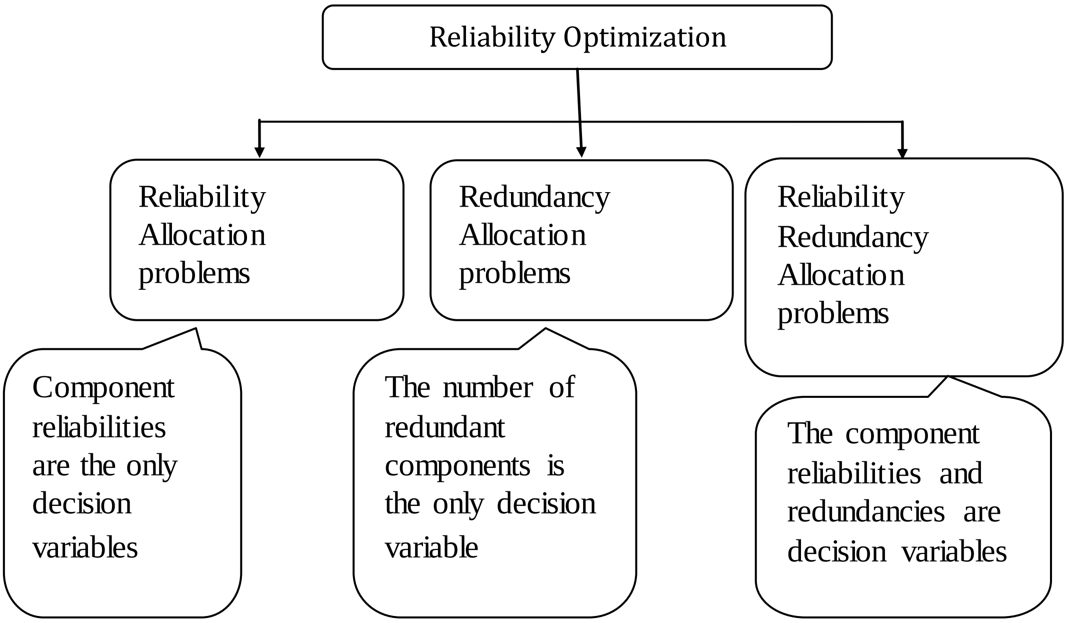

Reliability Optimization
Reliability Allocation problems
Redundancy Allocation problems
Reliability Redundancy Allocation problems
Component reliabilities are the only decision variables
The number of redundant components is the only decision variable
The component reliabilities and redundancies are decision variables

Figure 2.1: Classification of Reliability Optimization according to decision variables

For reliability allocation problems, one may refer to the works of Allella, Chiodo and Lauria [168], Yalaoui, Chatelet and Chu [169] and Salzar, Rocco and Galvan [170]. Researchers like Kim and Yum [171], Coit and Smith [172,173], Prasad and Kuo [174], Liang and Smith [175], Ramirez-Marquez and Coit [176], Yun and Kim [177], Nourelfath and Nahas [178], You and Chen [179], Agarwal and Gupta [180], Coit and Konak [181], Ha and Kuo [182], Tian and Zuo [183], Liang and Chen [184], Nahas and Nourelfath [185], and others have solved redundancy allocation problem.

Also, Federowicz and Mazumdar [186], Tillman, Hwang and Kuo [187], Misra and Sharma [188], Dhingra [189], Painton and Campbell [190], Ha and Kuo [191,192], Chen [193], Kim, Bae and Park [194] and others have solved the reliability redundancy allocation problem.

Several researchers have considered standby redundancy such as Zhao and Liu [195], Zhao and Song [196], Yu, Yalaoui, Chatelet and Chu [197], Prasad and Kuo [198], Ramirez-Marquez and Coit [199], Meziane, Massim, Zeblah, Ghoraf and Rahil [200], Tian, Levitin and Zuo [201] and Li, Chen, Yi and Tao [202]. Different algorithms for the reliability redundancy allocation problem have been discussed by Caserta et al., Sahoo et al., Banerjee et al. and Misra et al. [203-208].

2.7 Literature survey on Wireless Sensor Network (WSN)

Grossglauser [209] added the idea about the dramatic improvement in capacity scaling due to mobility of WSN and Diggavi [210] described how one-dimensional mobility increases ad hoc wireless capacity, which is a noteworthy contribution in this field. Some researchers focused on the improvement of WSN architecture and the communications protocol development. In this context, we can indicate the contribution of papers [211-214] in the optimization on reliability improvement of WSN as well as power consumption optimization in wireless sensor networks. Ailawadhi [211] worked on mobility issues in hybrid ad-hoc wireless sensor networks. Shah [212] explained about the Modeling of three-tier architecture for sparse sensor networks. Kansal [213] is about Controlled mobility for sustainable wireless networks, and Tong [214] describes the idea behind Sensor networks with mobile agents. As it is known, power consumption is one of the major constraints by which the reliability of WSN can be measured, as described in Chakrabarti [215]. Our work is inspired by the work presented in Chakrabarti [215], which proposed the Communication Power Optimization in a Sensor Network with a Path-Constrained Mobile Observer, which is based on rigorous analysis and case studies with real datasets to solve the power optimization problem pertinent to mobile WSNs.

For details one may referred the works of Raghunathan et al.[216], Akyildiz et al. [217], Kim et al.[218], Tillman et al.[219], Dima et al.[220],Cinque et al.[221], Shrestha et al.[222], Shaikh et al. [223], Banerjee et al.[224], Parameswaran et al.[225]. Raghunathan Vijay et al. [216] added the idea about reliability improvement in WSN using efficient algorithms. Akyildiz et al. [217] has done an elaborated survey on wireless sensor network. Different techniques for finding reliability in WSN systems are described in the paper of Kim et al.[218]. Paper of Tillman et al.[219] described how an optimal allocation of reliability can be achieved in Complex (Bridge) Systems. A fuzzy knowledge-based approach has been tested in paper of S. M., et al. [220] with the aim to increase the reliability of WSN systems. Other reliability assessment methodologies are used in papers of Cinque et al. [221],Shrestha et al. [222], Shaikh et al. [223] for WSN planning or operation.

2.8 Literature survey on Power Distribution System (PDS)

System Average Interruption Frequency Index (SAIFI) and System Average Interruption Duration Index (SAIDI) are two such reliability indices which are directly connected to the performance (Billinton and Allan)[226] of a PDS. SAIFI is the “average number of interruptions that a customer would experience”. SAIDI is the “average outage duration for each customer served” (Pregelj et al.)[227]. There are other reliability metrics too such as MAIFI (“average number of momentary interruptions that a customer would experience during a given period, typically a year”) (Brown)[228].

Detailed information about widely used indices for quantitative assessment of reliability of a PDS (such as SAIFI, SAIDI and MAIFI) is presented in IEEE 1366 [229] standard. Billinton and Allan [230] recommend SAIFI and SAIDI indices as customer-oriented metrics that can reflect the significance of a system outage. Hilber and Bertling [231] have also clearly described the relationship between component reliability and its effect on the whole system. Some other reliability indices are also available in the literature but the main drawback of these indices is their high computing cost.

One of the most commonly used methods to improve the reliability of distribution systems uses the so-called Reliability Centered Maintenance (RCM) technique. The method described by Tirapong and Titti [232] is based on the analysis of electricity supply interruptions, maintenance activities, costs of these activities and scheduling them based on Reliability Improvement Opportunity Graph. Another RCM- type method proposed by Li and Brown [233] provides high-level reliability and low costs based on maintenance scheduling using the "benefit-to-cost ratio" criterion. The proposed method determines a priority list of maintenance activities based on component and system reliability. However, the assumption of task independency considered before applying the proposed method must be checked further for consistency. Ramírez-Rosado and Domínguez-Navarro [234] compute optimal sizes and locations of reserve feeders for maximizing the level of reliability at the minimum economic cost. Teng and Lu [235] attempt to improve service reliability and reduce customer outage costs by computing optimal locations of feeder sectionalizers. However, relocating feeders may employ higher cost than relocating protective devices and switches.

The issue of improving PDS reliability by optimal location of switching and protective devices is not a new approach. Soudi and Tomsovic [236] proposed a binary programming optimization to identify types and locations of protective devices on a distribution feeder to ensure reliable and low cost power supply. Teng and Liu [237] presented an Ant colony system (ACS) based algorithm for optimum switch relocation to reduce interruption cost. Da Silva et al. [238] presented a novel technique to optimally place both control and protective devices using Mixed Integer Non-Linear Programming (MINLP) with real and binary variables. Another approach for optimal placement of reclosers and distributed generators to enhance system reliability through minimization of reliability indices like SAIFI, SAIDI, and MAIFI was proposed by Popović et al. [239]. Their optimization approach is based on sensitivity analysis of the power flow equations and reliability indices calculation combined with a Genetic Algorithm (GA) optimization strategy. The study was conducted for a limited range of operating conditions, but more realistic scenarios with time varying load and uncertainties have not been considered. These techniques also employ a single objective optimization problem to minimize cost or reliability indices.

A unique model to achieve higher reliability and lower cost by a multi-objective Ant Colony Optimization (ACO) technique was proposed by Tippachon and Rerkpreedapong [240]. This approach aims at minimizing the total cost and other two reliability indices (SAIFI and SAIDI) simultaneously. The results of the optimization process are optimal locations of switches and protective devices in a test PDS.

The method proposed by Ray et al. [241] aims at locating a set of remote control switches (RCS) optimally using Differential Search-type algorithm by applying a multi-objective function in order to improve system reliability at a low cost. However, their tests used fully reliable RCS and authors themselves stressed the need for a more realistic approach that takes failure rates and repair time for RCS into account.

2.9 Literature survey on Hybrid GA-PSO and GA-ACO

Heuristic optimization provides a robust and efficient approach for solving complex real-world problems. a hybrid method combining two heuristic optimization techniques, genetic algorithms(GA) and particle swarm optimization (PSO), for the global optimization of multimodal functions has been proposed by Kao et al. [242] . Saravanan et al. [243] proposed an improved model approach to single diode PV model by Hybrid Genetic Algorithm - Particle Swarm Optimization (Hybrid GA-PSO) technique. The main objective of the research work is to extract accurate parameters of PV model. Jeong et al. [244] has been developed a sophisticated GA/PSO-hybrid algorithm for application to real-world optimization problems was proposed. This new hybrid algorithm has been applied to two-test-function problems.

Electricity load forecasting is a challenging task because electric load has complex and nonlinear relationships with several factors. Sheikhan et al. [245] two hybrid models are developed for short-term load forecasting (STLF). These models use ant colony optimization and genetic algorithm. Zukhri et al. [246] proposed an optimization problem based on Genetic Algorithm (GA) and Ant Colony Optimization Algorithm (ACO). GA is designed by adopting the natural evolution process, while ACO is inspired by the foraging behaviour of ant species. In this paper presents a hybrid GA-ACO for Travelling Salesman Problem (TSP), called Genetic Ant Colony Optimization (GACO). Fidanova et al. [247] proposed a hybrid scheme, to solve optimization problems, using a Genetic Algorithm (GA) and an Ant Colony Optimization (ACO). In the hybrid GA-ACO approach, the GA is used to find a feasible solution to the considered optimization problem. Next, the ACO exploits the information gathered by the GA.

2.10 Major observations and scope

Observations and scope from literature survey of Evolutionary Algorithms (EA)

There are some scope of modification of operators of evolutionary algorithms and unique strategy design. There are some scopes of EA model design in different Engineering applications to increase reliability. Usage of EA models may be focused in specific engineering fields like Wireless Sensor Network (WSN) and Power Distribution System (PDS).

Observations and scope from literature survey of Genetic Algorithms (GA)

Through study it has been observed that there is some necessity of effective algorithms in the field of reliability optimization. There are some scopes of designing efficient Genetic Algorithm by the modification of different GA operators like crossover operator and mutation operator.. There are also some scopes of applying GA in different Engineering applications to increase reliability.

Observations and scope from literature survey of particle Swarm optimization (PSO)

By through study it has been understood that there are immense scope of modification of different operator in PSO. Cost effectiveness and reliability optimization problem can be solved in case of different Engineering fields like Wireless Sensor Network (WSN).

Observations and scope from literature survey of Differential Evolution (DE)

Modification in different operators on efficient DE approach can be done. Cost effectiveness and enhancement on reliability may be focused using the modified Differential Evolution algorithm. There is enormous scope of applying DE algorithm in different Engineering applications to increase reliability.

Observations and scope from literature survey of Ant Colony Optimization (ACO)

It has been observed through trough study that the modification in different operators on new efficient ACO approach can lead to a good research work to reliability enhancement of different engineering fields.

Observations and scope from literature survey of GA-PSO hybrid algorithm

It has been realized that there is huge scope of applying modern and efficient GA-PSO hybrid algorithm to solve reliability redundancy allocation problem in the field of Wireless Sensor Network (WSN). It is also understood that there is immense scope of GA-PSO in different Engineering applications to increase reliability.

Observations and scope from literature survey of GA-ACO hybrid algorithm

Through study it has been understood that there is a huge scope of applying GA-ACO hybrid algorithm in different engineering fields like Power distribution system of power engineering to solve reliability redundancy allocation problem.

CHAPTER 3

Solution Methodology

3.1 Introduction

Over the last few decades, several researchers formulated and solved single objective optimization problems and/or multi-objective optimization problems as integer non-linear programming problems (INLPP) and/or mixed-integer non-linear programming problems (MINLPP) with single or several resource constraints. To solve those problems, they proposed different techniques. In this regard, one may refer to the works of Tillman, Hwang and Kuo [248-250], Nakagawa, Nakashima and Hattori [251], Misra and Sharma [252], Chern [253], Ohtagaki et al. [254], Kuo et al. [255], Sun and Li [256], Gen and Yun [257], Ha and Kuo [258] , Coelho [259,260] among others. In their works, the design parameters involved in optimization problems have been considered as precise values. This means that every probability involved in the problem is perfectly determinable. In this case, it is usually assumed that complete probabilistic information about the system and its components' behaviour is available. However, in real-life situations, there is not sufficient statistical data available in most of the cases where the system is either new or it exists only as a project. It is not always possible to observe the constancy from the statistical point of view. This means that only partial information about the parameters is known. In these cases, the parameters are said to be imprecise. To tackle a problem with such imprecise parameters, usually stochastic, fuzzy, interval and fuzzy-stochastic approaches are applied and the corresponding problems are converted into deterministic problems for solving them.

3.2 Some Mathematical Background

3.2.1 Interval Number

An interval number A is a closed subset of $\mathbb{R}$ denoted by $A=[\underline{a},\overline{a}]$ and is defined by $A=[\underline{a},\overline{a}]=\{x:\underline{a}\le x\le\overline{a},x\in\mathbb{R}\}$, where $\underline{a}$ and $\overline{a}$ are the lower and upper bounds respectively and $\mathbb{R}$ is the set of all real numbers.

Every real number can also be treated as an interval, such as for all $x\in\mathbb{R}$, x can be written as an interval $[x,x]$.

Here, we shall give some basic arithmetical operations like addition, subtraction, multiplication and division of interval numbers.

Let $A=[\underline{a},\overline{a}]$ and $B=[\underline{b},\overline{b}]$ be two intervals.

Then the addition of two intervals A and B is given by

$$A+B=[\underline{a}+\underline{b},\overline{a}+\overline{b}] \tag{3.1}$$

The subtraction of two intervals A and B is given by

$$A-B=[\underline{a}-\overline{b},\overline{a}-\underline{b}] \tag{3.2}$$

The multiplication of an interval A by a real number θ is defined by

$$\theta A=\begin{cases}[\theta\underline{a},\theta\overline{a}] & \text{for}\,\theta\ge 0,\\ [\theta\overline{a},\theta\underline{a}] & \text{for}\,\theta<0,\end{cases} \tag{3.3}$$

The mid-point of an interval A is denoted by $m(A)$ and is defined by

$$m(A)=\frac{\underline{a}+\overline{a}}{2} \tag{3.4}$$

The product of two different intervals A and B is defined by

$$A\times B=[\min(\underline{a}\underline{b},\underline{a}\overline{b},\overline{a}\underline{b},\overline{a}\overline{b}),\max(\underline{a}\underline{b},\underline{a}\overline{b},\overline{a}\underline{b},\overline{a}\overline{b})] \tag{3.5}$$

The division of the interval B by the interval A is defined as

$$\frac{B}{A}=B\times\frac{1}{A}=[\underline{b},\overline{b}]\times[\frac{1}{\overline{a}},\frac{1}{\underline{a}}],\ \text{provided}\,0\notin[\underline{a},\overline{a}] \tag{3.6}$$

The above definitions are provided in the books written by Moore [261] and Hansen and Walster [262].

3.2.2 Fuzzy Sets

3.2.2.1 Fuzzy Set: Let X be a non-empty set. Then a fuzzy set $\tilde{A}$ in X is a set of ordered pair given by $\tilde{A}=\{(x,\mu_{\tilde{A}}(x)):x\in X\}$ where $\mu_{\tilde{A}}(x):X\rightarrow[0,1]$ is a function such that $0\leq\mu_{\tilde{A}}(x)\leq 1\forall x\in X$, and $\mu_{\tilde{A}}(x)$ represents the grade of membership of x in $\tilde{A}$.

3.2.2.2 α-Level Set: Let $\alpha\in[0,1]$, then α - level set or α –cut of a fuzzy set generated by fuzzy set $\tilde{A}$ is denoted by $\tilde{A}_{\alpha}$ and defined by $\tilde{A}_{\alpha}=\{x\in X:\mu_{\tilde{A}}(x)\geq\alpha\}$.

3.2.2.3 Normal Fuzzy Set: A fuzzy set $\tilde{A}$ is called a normal fuzzy set if there exists at least one $x\in X$ such that $\mu_{\tilde{A}}(x)=1$.

3.2.2.4 Convex Fuzzy Set: A fuzzy set $\tilde{A}$ is called convex iff for every pair of $x_1,x_2\in X$, the membership function of $\tilde{A}$ satisfies the inequality $\mu_{\tilde{A}}(\theta x_1+(1-\theta)x_2\geq\min\{\mu_{\tilde{A}}(x_1),\mu_{\tilde{A}}(x_2)\}$, where $\theta\in[0,1]$.

3.2.2.5 A Fuzzy Number: A fuzzy number $\tilde{A}$ is a fuzzy set which is both convex and normal.

3.2.2.5.1 Triangular Fuzzy Number (TFN)

A triangular fuzzy number $\tilde{A}$ is represented by (a_1,a_2,a_3) and defined by its membership function $\mu_{\tilde{A}}(x):X\rightarrow[0,1]$ given by

$$\mu_{\tilde{A}}(x)=\begin{cases}\dfrac{x-a_1}{a_1-a_2}\ when\,a_1\leq x<a_2\\ 1\ \ when\ \ x=a_2\\ \dfrac{a_3-x}{a_3-a_2}when\,a_2<x\leq a_3\end{cases} \tag{3.7}$$

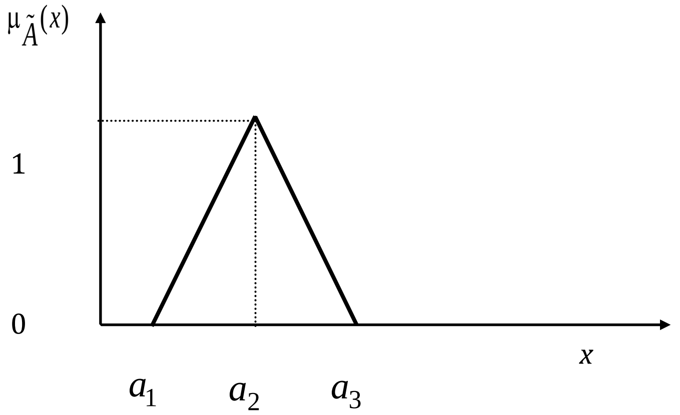


Figure 3.1: Pictorial representation of triangular fuzzy number

3.2.2.5.2 Parabolic Fuzzy Number (PFN)

A parabolic fuzzy number $\tilde{A}$ is represented by (a_1, a_2, a_3) and defined by its continuous membership function $\mu_{\tilde{A}}(x) : X \to [0,1]$ given by

$$\mu_{\tilde{A}}(x) = \begin{cases} 1 - \left(\dfrac{a_2 - x}{a_1 - a_2} \right)^2 & when\, a_1 \le x < a_2 \\ 1 & when \;\; x = a_2 \\ 1 - \left(\dfrac{x - a_2}{a_3 - a_2} \right)^2 & when\, a_2 < x \le a_3 \end{cases} \tag{3.8}$$

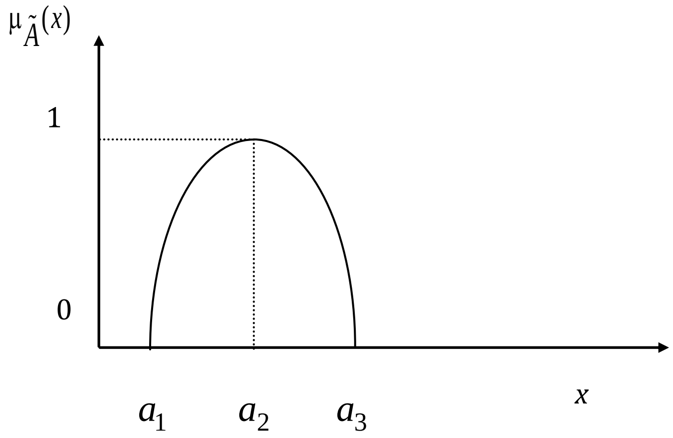


Figure 3.2: Pictorial representation of Parabolic fuzzy number

3.2.2.5.3 Trapezoidal Fuzzy Number (TrFN)

A trapezoidal fuzzy number $\tilde{A}$ is represented by (a_1, a_2, a_3, a_4) and defined by its membership function $\mu_{\tilde{A}}(x) : X \to [0,1]$ given by

$$\mu_{\tilde{A}}(x) = \begin{cases} \dfrac{x - a_1}{a_2 - a_1} & \text{if } \; a_1 \le x \le a_2 \\ 1 \text{ if } \; a_2 \le x \le a_3 \\ \dfrac{a_4 - x}{a_4 - a_3} & \text{if } \; a_3 \le x \le a_4 \end{cases} \tag{3.9}$$

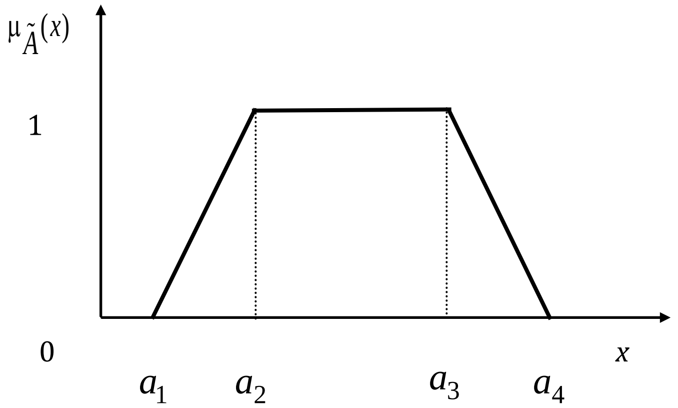


Figure 3.3: Pictorial representation of trapezoidal fuzzy number

3.2.2.6 Defuzzification

Defuzzification is the process of producing a representable crisp value in fuzzy logic/fuzzy set, given the fuzzy sets and the corresponding degrees of membership. There are different types of defuzzification method available in the existing literature. However, some useful defuzzification methods are as follows:

(i) Centre of Area (COA) or Centre of Gravity (COG) or Centroid

(ii) Bisector of Area (BOA)

(iii)Smallest of Maxima (SOM)

(iv) Largest of Maxima (LOM)

(v)Mean of Maxima (MOM)

(vi) Regular Weighted Point (RWP)

3.2.2.6.1 Centre of Area (COA) or Centre of Gravity (COG) or Centroid

In this defuzzification method the centroid or center of gravity (COG) of the area under the membership function is calculated by following formula.

$$x_{COA} = \frac{\int_x x\mu_{\tilde{A}}(x)dx}{\int_x \mu_{\tilde{A}}(x)dx} \tag{3.10}$$

where $\mu_{\tilde{A}}(x)$ is the membership function and x is the output variable.

3.2.2.6.2 Bisector of Area (BOA)

This defuzzification can be expressed as

$$\int_{a_1}^{x_{BOA}} \mu_{\tilde{A}}(x)dx = \int_{x_{BOA}}^{a_4} \mu_{\tilde{A}}(x)dx \tag{3.11}$$

3.2.2.6.3 Smallest of Maxima (SOM)

It is the smallest value with maximum membership function. It is denoted as x_{SOM}.

3.2.2.6.4 Largest of Maxima (LOM)

It is the largest value among all x and $x \in [a_2, a_3]$.It is denoted as x_{LOM}.

3.2.2.6.5 Mean of Maxima (MOM)

It is the mean value of x_{SOM} and x_{LOM}.

It is defined by $x_{MOM} = \frac{x_{LOM} + x_{SOM}}{2}$ (3.12)

3.2.2.6.6 Regular Weighted Point (RWP)

For the fuzzy number $\tilde{A}$ the α – cut is $\tilde{A}_\alpha = [\underline{A}(\alpha), \bar{A}(\alpha)]$ and the regular weighted point for $\tilde{A}$ is given by

$$RWP(\tilde{A}) = \frac{\int_0^1 \left(\frac{\underline{A}(\alpha) + \bar{A}(\alpha)}{2}\right) f(\alpha) d\alpha}{\int_0^1 f(\alpha) d\alpha} \lim_{x \to \infty} \tag{3.13}$$

where $f(\alpha) = \begin{cases} 1 - 2\alpha \; when \; \alpha \in [0,1/2] \\ 2\alpha - 1 when \; \alpha \in [0,1/2] \end{cases}$

Formula for defuzzification of Trapezoidal fuzzy number using centroid method

The defuzzified value x_{COA} of trapezoid fuzzy number $\tilde{A} = (a_1, a_2, a_3, a_4)$ is given by

$$x_{COA} = \frac{a_3^2 + a_4^2 + a_3 a_4 - a_1^2 - a_2^2 - a_1 a_2}{3(a_3 + a_4 - a_1 - a_2)} \tag{3.14}$$

3.2.2.7 Fuzzy linguistic variable

A fuzzy set is a linguistic variable when it is represented by descriptive words in natural languages. The concept of linguistic variable provides a means of approximate characterization of phenomena which are too complex or too ill-defined to describe in conventional quantitative terms. Linguistic values can be represented using fuzzy numbers. For example “cost” is a linguistic variable if its values are linguistic rather than numerical i.e., “small”, “medium” and “high”. Here, linguistic variable is represented by using rating set. The rating set is defined as $RS = \{S, M, H\}$ where, S = Small, M = Medium and H=High.

3.3 Solution Methodologies

3.3.1 Genetic Algorithms (GA)

Genetic Algorithm (GA) is a computerized stochastic search optimization technique [263]. Gen and Cheng [264] described the application of GA to several combinatorial problems including reliability optimization problems. GA is the most powerful and widely known evolutionary computational algorithm due to its simplicity, effectiveness and wide range of applications. It works by the evolutionary principles [265] and chromosomal processing in natural genetics. The flowchart of GA has been shown in figure 3.4.

GA Terminology

It is very important to understand the terminology that has been used in genetic algorithm. Some of the commonly used terms are as follows:

Population: A collection of several alternative solutions to the given problem is called a population.

Chromosome: Each individual in the population is called a chromosome.

Genes: Often these individuals are coded as binary/real strings and the individual character or symbol in the string is called as genes.

Fitness Function: It is an evolution function, which is used to determine the fitness of each chromosome. The fitness function is usually user defined and problem specific.

Solution Space: The range of possible solutions is referred to as the solution space and the fitness of each point is referred to as the altitude in the landscape of the problem.

Generation Gap: It is the fraction of the individuals in the population that are replaced from one generation to the next and is equal to one for simple GA.

Termination Criterion:

The termination criterion is a condition for which the algorithm/process is going to stop. For this purpose any one of the following conditions is considered as the termination criterion.

(i) The best individual does not improve over specified generations.

(ii) The total improvement of the last certain number of best solutions is less than a pre-assigned small positive number.

(iii) The number of generations reaches a prescribed finite number of generation (called maximum number of generations).

The following vital components have been taken into account for implementation of genetic algorithm:

(i) GA parameters

(ii) Chromosome representation

(iii) Initialization of population

(iv) Evaluation of fitness function

(v) Selection process

(vi) Genetic operators (crossover and mutation)

GA parameters

Population size, maximum number of generations, crossover rate and mutation rate are four important parameters in GA. The values of those parameters are chosen with an intension to get better optimal result which is closely nearest to global optima.

Chromosome representation

Representation of chromosome is an important task in any meta-heuristic/heuristic algorithm. There are different types of representation used to represent the chromosome. Here we have used real coding for representing the chromosomes.

Evaluation of fitness function

Evaluation/fitness function plays an important role in GA. This role is same for natural evolution process in the biological environments. After initialization of chromosomes of potential solutions, we need to see how relatively best they are. Therefore, we have to calculate the fitness value for each chromosome.

GA Operators

Here we shall discuss only three main components of GA viz. selection process, crossover and mutation.

In GA, the selection operator plays an important role as it is the first operator applied to the population. The aim of this operator is to select the above average solutions and eliminate below average solutions from the entire population for the next generation under the principle “survival of the fittest”.

Selection Process

There are many selection schemes for GA, each with different characteristics. Among them tournament selection is a helpful and robust selection mechanism, commonly used by GA. This selection process is a refining process which selects the capable off-springs from the tournament batch. A high number of competitors in the tournament guaranty a higher selection pressure/stress. An ideal selection scheme would be simple to code and efficient for both parallel and non-parallel architectures. Furthermore, a selection scheme should be able to adjust its selection pressure so as to tune its performance for different domains.

In this thesis, we have used the tournament selection process of size two with replacement as the selection operator, with the following assumptions:

(i) When both the chromosomes are feasible, then the one with better fitness value is selected.

(ii) When one chromosome is feasible and another is infeasible, then the feasible one is selected.

(iii) When both the chromosomes are infeasible with unequal constraints violation, then the chromosome with less constraints violation is selected.

(iv) When both the chromosomes are infeasible with equal constraints violation, then any of the two chromosomes is selected.

After the selection process, a crossover operator is applied to the resulting chromosomes which have survived. It is an operation that really empowers the GA. It operates on two or more parent chromosomes at a time and creates the offspring by recombining the features of the parent solutions. In this thesis, we have used intermediate crossover for integer variables. For floating point variables, we have proposed a new crossover scheme and we named it power crossover.

The different steps of crossover operation are as follows:

Step-1: *Find the integral value of* $\lfloor P_c * P_s \rfloor$ *and store it in* N_c.

Step-2: *Select two parent chromosomes* $s_k^{(t)}$ *and* $s_i^{(t)}$ *randomly from the population.*

Step-3: *Compute the two offspring components* $\bar{s}_{kj}^{(t)}$ *and* $\bar{s}_{ij}^{(t)}$ $(j=1,2,\dots,n)$ *from the parent chromosomes* $s_k^{(t)}$ *and* $s_i^{(t)}$.

Now, we discuss the above minimized crossover operators viz., Intermediate crossover and power crossover.

Intermediate crossover:

In the case of intermediate crossover, the two offspring components $\bar{s}_{kj}^{(t)}$ and $\bar{s}_{ij}^{(t)}$ $(j=1,2,\dots,n)$ will be created by $\bar{s}_{kj}^{(t)} = s_{kj}^{(t)} - g$ and $\bar{s}_{ij}^{(t)} = s_{ij}^{(t)} + g$ if $s_{kj}^{(t)} > s_{ij}^{(t)}$, otherwise $\bar{s}_{kj}^{(t)} = s_{kj}^{(t)} + g$ and $\bar{s}_{ij}^{(t)} = s_{ij}^{(t)} - g$, where g is a random integer number between 0 and $\left|s_{kj}^{(t)} - s_{ij}^{(t)}\right|$, $j=1,2,\dots,n$

Power crossover:

In case of power crossover, the two offspring components $\bar{s}_{kj}$ and $\bar{s}_{ij}$ $(j=1,2,\dots,n)$ will be created by

$$\bar{s}_{kj}^{(t)} = \left(s_{kj}^{(t)}\right)^{r} \left(s_{ij}^{(t)}\right)^{1-r} \tag{3.15}$$

$$\bar{s}_{ij}^{(t)} = \left(s_{kj}^{(t)}\right)^{1-r} \left(s_{ij}^{(t)}\right)^{r} \tag{3.16}$$

where r is a random number uniformly distributed in the interval (0,1).

Step-4: Compute $s_k^{(t+1)} = \text{argument of best of } \{ f(s_k^{(t)}), f(s_i^{(t)}), f(\bar{s}_k^{(t)}), f(\bar{s}_i^{(t)})\}$

and $s_i^{(t+1)} = \text{argument of best of } \{ f(s_k^{(t)}), f(s_i^{(t)}), f(\bar{s}_k^{(t)}), f(\bar{s}_i^{(t)})\}$

where $f(.)$ denotes the fitness function.

Step-5: Repeat *Step-2-4 for* $\frac{N_c}{2}$ *times.*

The aim of the mutation operation is to introduce random changes into the population, in order to prevent the search process from converging to the local optima. Sometimes, it helps to regain information lost in earlier generations and it is responsible for the fine tuning of the system. This operator is applied to a single chromosome only. Usually, its rate is very low; because otherwise it would significantly alter the solutions generated through the selection and crossover operations. In this thesis, we have used one-neighbourhood mutation for integer variables proposed by Bhunia, Sahoo and Roy [266] and non-uniform mutation for floating point variables.

The different steps of mutations operations are as follows:

Step-1: *Find the integral value of* $\left[P_m * P_s * n\right]$ *and store it in* N_m.

Step-2: *Select a particular gene* $s_{ik}^{(t)}\,(k=1,2,\ldots,m)$ *on chromosome* $s_i^{(t)}$ *for mutation and domain of* $s_{ik}^{(t)}$ *is* $[l_{ik}, u_{ik}]$.

Step-3: *Create new gene* $s_{ik}'^{(t)}\;(k=1,2,\ldots,n)$ *corresponding to the selected gene* $s_{ik}^{(t)}$ *by mutation process.*

Now, we discuss the above minimized crossover operators viz., One-neighbourhood mutation and Non-uniform mutation.

One-neighbourhood mutation:

In case of one-neighbourhood mutation,

$$s_{ik}'^{(t)} = \begin{cases} s_{ik}^{(t)}+1 & \text{if } s_{ik}^{(t)} = l_{ik} \\ s_{ik}^{(t)}-1 & \text{if } s_{ik}^{(t)} = u_{ik} \\ s_{ik}^{(t)}+1 & \text{if } r < 0.5 \\ s_{ik}^{(t)}-1 & \text{if } r \geq 0.5 \end{cases} \tag{3.17}$$

where r random number uniformly distributed in (0, 1).

Non-uniform mutation:

In case of non-uniform mutation,

$$s_{ik}'^{(t)} = s_{ik}^{(t)} + \left(s_{il}^{(t)} - s_{ir}^{(t)}\right) * r_1 * (1 - t/M_g)^2 \;\; when\, r_2 < 0.5$$

otherwise $s_{ik}'^{(t)} = s_{ik}^{(t)} + \left(s_{il}^{(t)} - s_{ir}^{(t)}\right) * r_2 * (1 - t/M_g)^2$ (3.18)

where r_1 and r_2 are two random numbers uniformly distributed in (0, 1).

Step-4: Compute $s_i^{(t+1)} = \text{argument of better of } \{\, f(s_i^{(t)}),\, f(s_i'^{(t)})\}$

Step-5: Repeat Step-2 to Step-4 for N_m times.

Algorithm for GA

Step 1: Set population size (P_s), maximum number of generations (M_g), probability of crossover (P_c), probability of mutation (P_m) and the bounds of decision variables.

Step 2: Set t=0.

Step 3: Initialize the chromosomes of the population $P(t)$.

Step 4: Compute the fitness function for each chromosome of $P(t)$.

Step 5: Find the chromosome having the best fitness value.

Step 6: Set *t=t +1*.

Step 7: If the termination condition is satisfied, then go to Step-13; otherwise go to the next step.

Step 8: Select the population $P(t)$ from the population $P(t-1)$ of (*t*-1)- th generation using the selection operator.

Step 9: Apply the crossover and mutation operators on $P(t)$ to produce new population members $P(t)$.

Step 10: Compute the fitness function value of each chromosome of $P(t)$.

Step 11: Find the best chromosome from $P(t)$.

Step 12: Find the better of the best chromosomes of $P(t)$ and $P(t-1)$ store it; go to Step-6.

Step 13: Print the best chromosome and its fitness value.

Step 14: End.

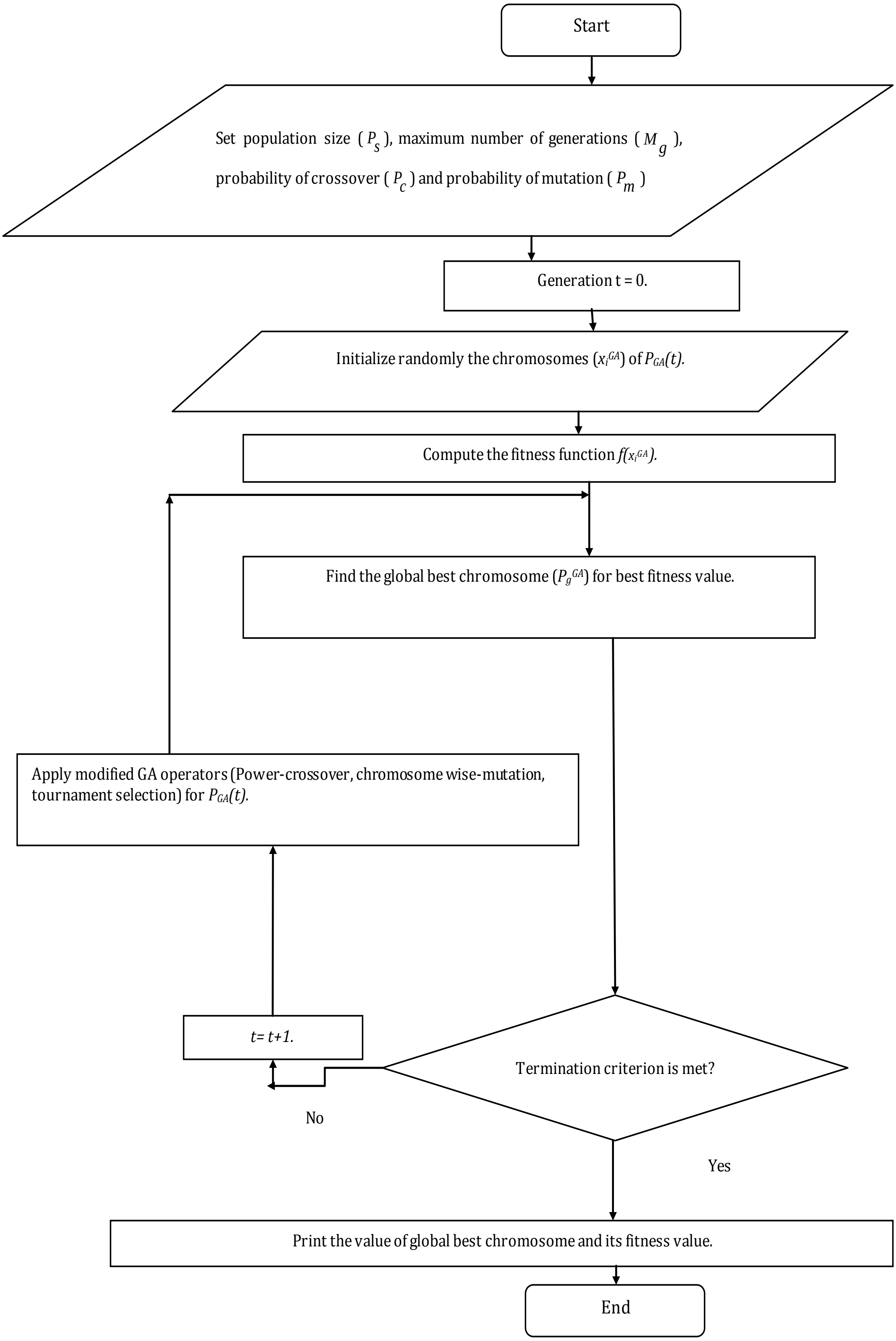

Start
Set population size (P_s), maximum number of generations (M_g), probability of crossover (P_c) and probability of mutation (P_m)
Generation t = 0.
Initialize randomly the chromosomes (x_i^{GA}) of $P_{GA}(t)$.
Compute the fitness function $f(x_i^{GA})$.
Find the global best chromosome (P_g^{GA}) for best fitness value.
Apply modified GA operators (Power-crossover, chromosome wise-mutation, tournament selection) for $P_{GA}(t)$.
$t= t+1$.
Termination criterion is met?
No
Yes
Print the value of global best chromosome and its fitness value.
End

Figure 3.4: Flowchart of the Genetic Algorithm (GA)

3.3.2 Particle Swarm Optimization (PSO)

Like Genetic Algorithms, Particle Swarm Optimization (PSO) is another meta-heuristic algorithm. It was originally developed by Eberhart and Kennedy in 1995 [267] and Kennedy in 2011 [268]. It is a population based heuristic global search algorithm based on social interaction and individual experience. PSO is an optimization tool based on a population where each member is considered as a particle and each particle is a potential solution of the optimization problem. PSO has a randomized velocity associated to it, which moves particles through the space of the problem. However, unlike Genetic Algorithms, PSO does not have operators such as crossover and mutation. In PSO, instead of the well-known evolutionary principle 'survival of the fittest', the simulation of social behaviour is much used.

Let P_s denote the swarm size and *n*, the dimensionality of the search space. Each particle *i* ($1 \le i \le P_s$) has the following attributes:

(i) A current position $x_i = (x_{i1}, x_{i2}, \ldots\ldots, x_{in})$ in the search spaces.

(ii) A current velocity $v_i = (v_{i1}, v_{i2}, \ldots\ldots, v_{in})$

(iii) A personal best (pbest) position (the position giving the best fitness value experienced by the particle) $p_i = (p_{i1}, p_{i2}, \ldots\ldots, p_{in})$.

At each iteration, the velocity of each particle in the swarm is updated as follows:

$$v_{ij}^{(k+1)} = wv_{ij}^{(k)} + c_1 r_{1j}\left(p_{ij}^{(k)} - x_{ij}^{(k)}\right) + c_2 r_{2j}\left(p_{gj}^{(k)} - x_{ij}^{(k)}\right),\ j = 1,2,\ldots,n;$$

$$k = 1,2,\ldots,n \qquad (3.19)$$

i.e., $$v_i^{(k+1)} = wv_i^{(k)} + c_1 r_1\left(p_i^{(k)} - x_i^{(k)}\right) + c_2 r_2\left(p_g^{(k)} - x_i^{(k)}\right) \qquad (3.20)$$

where $v_{ij}^{(k)}$ is the *j*-th component of velocity of *i*-th particle in *k*-th iteration, *w* is the inertia weight, *c1* and *c2* be called the acceleration coefficients, $r_{1j}^{(k)}$, $r_{2j}^{(k)}$ are two random numbers uniformly distributed in the interval (0,1) i.e., $r_{1j}^{(k)} \in U(0,1)$ and $r_{2j}^{(k)} \in U(0,1)$.

The new position of the *i*-th particle is computed as follows:

$$x_{ij}^{(k+1)} = x_{ij}^{(k)} + v_{ij}^{(k+1)} \quad \text{i.e.,} \quad x_i^{(k+1)} = x_i^{(k)} + v_i^{(k+1)} \qquad (3.21)$$

The personal best (pbest) position of each particle is updated as follows:

$$p_i^{(0)} = x_i^{(0)} \tag{3.22}$$

$$p_i^{(k+1)} = \begin{cases} p_i^{(k)} & \text{if } f(x_i^{(k+1)}) \le f(p_i^{(k)}) \\ x_i^{(k)} & \text{if } f(x_i^{(k+1)}) > f(p_i^{(k)}) \end{cases} \tag{3.23}$$

where the fitness function *f* is to be maximized.

The global best (gbest) position found by any particle during all previous iterations p_g is defined as $p_g^{(k+1)} = \arg \underset{p_i}{\max} \, f(p_i^{(k+1)}), \;\; 1 \le i \le P_s$ (3.24)

From the above equation, it is seen that the velocity of the *i*-th particle is computed by considering the following three components:

(i) The previous velocity of the particle

(ii) The distance between the particle's best previous and current positions

(iii) The distance between the swarm's best experience (the position of the best particle in the swarm) and the current position of the particle.

The velocity parameter in above equations are also bounded by the range $[-v_{\max}, v_{\max}]$ where $v_{\max}$ is called the maximum velocity of the particle. The choice of a small value for the maximum velocity ($v_{\max}$) can cause very small updating of velocities and positions of particles at each iteration. Hence, the algorithm may take a long computational time to converge. To overcome this difficulty, Clerc [269] and Clerc and Kennedy [270] proposed an improved velocity update rule considering a constriction factor χ . According to Clerc and Kennedy [270], the updated velocity is given by

$$v_i^{(k+1)} = \chi \left[v_i^{(k)} + c_1 r_1 \left(p_i^{(k)} - x_i^{(k)} \right) + c_2 r_2 \left(p_g^{(k)} - x_i^{(k)} \right) \right] \tag{3.25}$$

The constriction factor χ used in above expression is expressed as

$$\chi = \frac{2}{\left| 2 - \phi - \sqrt{\phi^2 - 4\phi} \right|} \tag{3.26}$$

where $\phi = c_1 + c_2$, $\phi > 4$ and χ is a function of c_1 and c_2. Generally, c_1 and c_2 are both set to be 2.05. Hence, ϕ is set to 4.1 and the value of χ is 0.727. This PSO is known as PSO-Co, i.e., constriction coefficient-based PSO [271].

Quantum behaved PSO (QPSO)

Sun et al. [272] first introduced this algorithm. It holds the basic behavior of PSO, but some modified operators involved here make it different from the traditional PSO. We can say that the traditional PSO follows a Newtonian approach to describe the movement of the particles, whereas QPSO considers the quantum behavior of particles, based on the principle of quantum mechanics.

In quantum mechanics, the time-dependent Schrödinger equation is as follows.

$$i\hbar\frac{\partial}{\partial t}\Psi(r,t) = H(r)\Psi(r,t) \tag{3.27}$$

where, i is the imaginary number i.e., $\sqrt{-1}$, $H(r) = -\frac{\hbar^2}{2m}\tilde{N}^2 + V(r)$, the time-dependent Hamiltonian operator; $\hbar$, the Planck's constant; *m,* the mass of the particle and $V(r)$, a potential energy distribution function, influencing the potential energy of the particle.

The squared amplitude $Q = |\Psi|^2$, i.e., square of the wave function $\Psi(r,t)$ in the above equation serves as a probability measure for the movement of the particle.

Now, under normalization, we can write

$$\iiint |\Psi(\mathrm{r},\mathrm{t})|^2 dxdydz = 1.0 \tag{3.28}$$

In QPSO, the swarm is considered as a quantum system where each particle has a quantum state based on the wave function as follows:

$$\tilde{p}_{ij} = \frac{\varphi_1 p_{ij} + \varphi_2 p_{gj}}{\varphi_1 + \varphi_2},\ i, j = 1, 2, \ldots, n \tag{3.29}$$

where, p_i is the best previous position (the position giving the best fitness value) of particle *i.* p_j denotes the position of the j-th particle after position update , p_g is the overall best position of the swarm-and $\varphi_1 = c_1 r_1$ and $\varphi_2 = c_2 r_2$, with c_1, c_2 are the cognitive and social parameters of PSO respectively and r_1, r_2 being uniformly distributed random numbers in [0,1].

Now, to describe the principle of QPSO, we are going to consider the simplest one dimensional case. Assuming that the center of the potential p_{ij} is defined by the above equation, according to Sun et al. [272] the potential energy distribution function is defined as:

$$V(x) = -\gamma\delta(p-x) = -\gamma\delta(y) \qquad (3.30)$$

where $y = p - x$, δ is the delta function of simple potential energy well and γ is a constant.

Through proper mathematical manipulations [272], we obtain the following wave function:

$$\Psi(y) = \frac{1}{\sqrt{L}} \exp(-\frac{|y|}{L}) \qquad (3.31)$$

where $L = \frac{\hbar^2}{m\gamma}$ is the characteristic length of potential well.

Hence the probability measure is as follows:

$$Q(y) = |\Psi(y)|^2 = \frac{1}{L} \exp(-2\frac{|y|}{L}) \qquad (3.32)$$

So far, we obtained a probability density of the particle positions. However, this is not adequate to serve as an algorithm, since the evaluation of a particle requires an exact position. Therefore, the position of the particle shall be estimated and this procedure is called collapse of the quantum state to the classical state. Collapsing is possible through a Monte Carlo simulation. More specifically, let s be a random number uniformly distributed in (0, $1/L$). Then, s can be written as $s = \frac{1}{L}u$, where u is a random number uniformly distributed in the range of [0, 1]. Substituting Q(y) in the left part of above equation with s, and solving for x, results in the following QPSO solution [272]:

$$x = p \pm \frac{L}{2} \ln(\frac{1}{u}) \qquad (3.33)$$

The above equation provides two possible new positions of the particle, which are measurable with the objective function. Sun et al. [272] provided a convergence proof of this QPSO model on the position p. The parameter L is the only control parameter that appears in the update equations of QPSO.

$$x_i(t) = p(t) + F(L, \pm u), \tag{3.34}$$

where, $p(t) = \dfrac{\varphi_1 p_i(t) + \varphi_2 p_g(t)}{\varphi_1 + \varphi_2}$

where i = 1, 2,..., N. u being a uniformly distributed number in the range of $[0,1]$, φ_1 and φ_2 being random numbers and F is a functional form obtained through the inversion of the probability density function, thereby depending on the employed quantum field model.

Algorithm for Quantum Particle Swarm Optimization (QPSO)

Step-1: Set Swarm as S, Size of swarm as N.

Step-2: Initialize swarm and personal best positions (*pbest*) and find the index of the best particle $p_g(t)$.

Step-3: Set $t = t$+1.

Step-4: If the termination condition is satisfied, then go to Step-6; otherwise go to the next step.

Step-5: Repeat the following for $i = 1, 2 \ldots, N$

(i) Compute the position $p(t)$ using $p_i(t)$ and $p_g(t)$.

(ii) Generate a random number $u \sim U(0,1)$.

(iii) Set $L = L(q, u, |x_i(t) - p(t)|)$. where q is quantum parameter and L is the initialization function of control parameter (L).

(iv) Generate a random number $R \sim U(0,1)$.

(v) Update the value of x_i by following the condition:

If $(R > 0.5)$ then $x_i(t+1) = x_i(t) + F(L,u)$ else $x_i(t+1) = x_i(t) + F(L,-u)$.

Step-6: Print the position and fitness of global best particle.

Step-7: End

As most of the PSO variants, the experiments on widely used test problems revealed that QPSO can become an efficient approach under proper fine-tuning.

The different philosophy than the rest of PSO variants, as well as its susceptibility of improvements [273,274] and its interesting applications [275,276] rendered QPSO a worth-mentioning approach.

Adaptive Quantum Particle Swarm Optimization (AQPSO)

The Quantum-behaved Particle Swarm Optimization algorithm (QPSO) with the phases of attraction and repulsion is called Adaptive Quantum-Behaved Particle Swarm Optimisation (AQPSO) algorithm and depending upon those phases the swarm nature is changed and diversity decreases [see equation no 3.36] and when diversity drops below a lower bound it switches to repulsion phase. The result of this is a QPSO algorithm that alternates between phases of exploiting and exploring attraction and repulsion-low diversity and high diversity ultimately the diversity reaches to the higher bound.

Algorithm for Adaptive Quantum Particle Swarm Optimization (AQPSO)

Step 1: Initialize the population taking a random variable x_i.

Step 2: If the termination condition is satisfied, go to step 3, otherwise follow the next sub-steps.

a) Find out the Mean Best Position $(mbest)$, which is defined as the set of center of gravity of global best (gbest) position of the particle as follows:

$$mbest = \left(\sum_{i=1}^{M} p_{i1} \Big/ M, \sum_{i=1}^{M} p_{i2} \Big/ M, \ldots, \sum_{i=1}^{M} p_{id} \Big/ M \right) \tag{3.35}$$

b) Measure the diversity of the swarm

$$diversity = \frac{1}{|S| . |L|} . \sum_{i=1}^{|S|} \sqrt{\sum_{j=1}^{n} (p_{ij} - \overline{p_j})^2} \tag{3.36}$$

c) If (diversity<dlow) then set $\beta = \beta_a$ i.e., Attraction Phase:

$\beta = \beta_a$ where $\beta_a \leq 1$

d) If (diversity>dhigh) then set $\beta = \beta_r$ i.e., Repulsion Phase:

$\beta = \beta_r$ where $\beta_a > 1$

e) Repeat the following for i=1 to M .

```
        Check if $(f(x_i) < f(p_i))$ then $p_i = x_i$ set $p_g = \min(p_i)$.
    f) Do the following for $d = 1$ to n
        (i) Set $\phi_1 = rand(0,1)$, $\phi_2 = rand(0,1)$ ,
            $p = (\phi_1 * p_{id} + \phi_2 * p_{gd}) / (\phi_1 + \phi_2)$ and $u = rand(0,1)$
        (ii) if $(rand(0,1) > 0.5)$
                $x_{id} = p - \beta * (mbest_d - x_{id}) * (\ln(1/u))$        (3.37)
             else
                $x_{id} = p + \beta * abs(mbest_d - x_{id}) * (\ln(1/u))$        (3.38)
Step 3: End
```

where, *S* is the swarm, *M=|S|,* the population size, *|L|,* the length of the longest diagonal in the search space*, n,* the dimensionality of the problem, p_{ij}, the j-th value of the i-th particle *(pbest)* and $\bar{p}_j$, the *j*-th value of the average point *p(mbest).* $\bar{p}_i = (p_{i1}, p_{i2}, ..., p_{in})$, is the best previous position (the position giving the best fitness value) of particle i. $\bar{p}_g = (p_{g1}, p_{g2}, ..., p_{gn})$, is the position of the best particle among all the particles in the population.

Algorithm for Weighted Quantum Particle Swarm Optimization (WQPSO)

In AQPSO algorithm, the mean best position is simply the average of the personal best position of all particles, which means that each particle is considered equal and exactly the same influence on the value of *m*, where *m* is defined as the mean of the *pbest* positions of all particles. The definition of the mainstream thought as mean of the personal best positions is somewhat reasonable. The greater the fitness, the more important the particle is. Describing it formally, we can rank the particles in descending order according to their fitness value. Then assign each particle a weight coefficient α_i linearly decreasing with the particle's rank, that is, the nearer the best solution, the larger its weight coefficient is. The mean best position *m*, therefore, is calculated as

$$m(t) = (m_1(t), m_2(t), \ldots, m_n(t)) = (\frac{1}{M}\sum_{i=1}^{M}\alpha_{i,1}P_{i,1}(t), \frac{1}{M}\sum_{i=1}^{M}\alpha_{i,2}P_{i,2}(t), \ldots, \frac{1}{M}\sum_{i=1}^{M}\alpha_{i,n}P_{i,n}(t)) \tag{3.39}$$

where α_i is the weight coefficient and $\alpha_{i,d}$, the dimension coefficient of every particle, M, the population size. In this chapter, the weight coefficient for each particle decreases linearly from 1.5 to 0.5. The improved algorithm is called Weighted QPSO that is given as follows.

Step 1: Initialize the population taking a random value of each variable x_i.

Step 2: If the termination condition is met, go to Step 3, otherwise perform the following sub-steps.

a) Find out the Mean Best Position $(mbest)$, which is defined as the set of center of gravity of global best (*gbest)* position of the particle using the above equation .

b) Repeat the following for i=1 to *M.*

Check if $(f(x_i) < f(p_i))$ then $p_i = x_i$, set $p_g =$ arg $\min(f\ (p_i\ )$

c) Repeat the following for $d = 1$ to n

i) Set $\eta = rand(0,1)$, $p_{ij} = \eta * p_{ij} + (1-\eta) * p_{gj}$ and

$u = rand(0,1)$

ii) if $(rand(0,1) > 0.5)$

$$x_{ij} = p_{ij} - \beta * abs(m_j - x_{ij}) * (\ln(1/u)) \tag{3.40}$$

else

$$x_{ij} = p_{ij} + \beta * abs(mbest_j - x_{ij}) * (\ln(1/u)) \tag{3.41}$$

Step 3: End

3.3.3 Differential Evolution (DE)

Differential Evolution (DE) was first introduced by Storn and Price in 1996 [277], in the ICSI technical report ("Differential Evolution — A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces", 1995). They observed that there are many practical problems having objective functions that are non-differential, non-linear and non-continuous or have several local minima, constraints and are stochastic in nature. Those problems are impossible to solve analytically and DE can give approximate solutions to such problems. DE is a population based, bio-inspired evolutionary algorithm. Storn and Price [277] detected that differential mutation combined with discrete recombination and pair wise selection does not need an annealing factor. So, they removed the annealing mechanism and thus the obtained algorithm started the era of differential evolution.

Now we can identify some properties of DE which make this algorithm special in comparison with other evolutionary algorithms. We can observe the following properties. At first, we have to specify that the global optimum can be attained easily through this algorithm. Secondly, this algorithm maintains a very good precision. Therefore, this algorithm is very much effective to get accurate results up to a higher level of approximation. The third, and one of the major properties of DE, is fast convergence, which makes this algorithm to obtain efficiently the global optimum. The fourth important property is self-adaptation, and through this property, this algorithm can easily be used in different stochastic environments, as well as in fuzzy environments also.

DE uses the differences between randomly selected vectors (individuals) as the source of random variations for a third vector (individual), referred to as the target vector. Trial solutions are generated by adding weighted difference vectors to the target vector. This procedure is referred to as the mutation operator where the target vector is mutated. A recombination (or crossover) step is then applied to produce an offspring which is only accepted if it improves on the fitness of the parent individual. Traditional DE is the combinations of four major operators and those are initialization of random variables, mutation, recombination or crossover and selection or optimum respectively. The block diagram of DE [278] has been shown in figure 3.5.

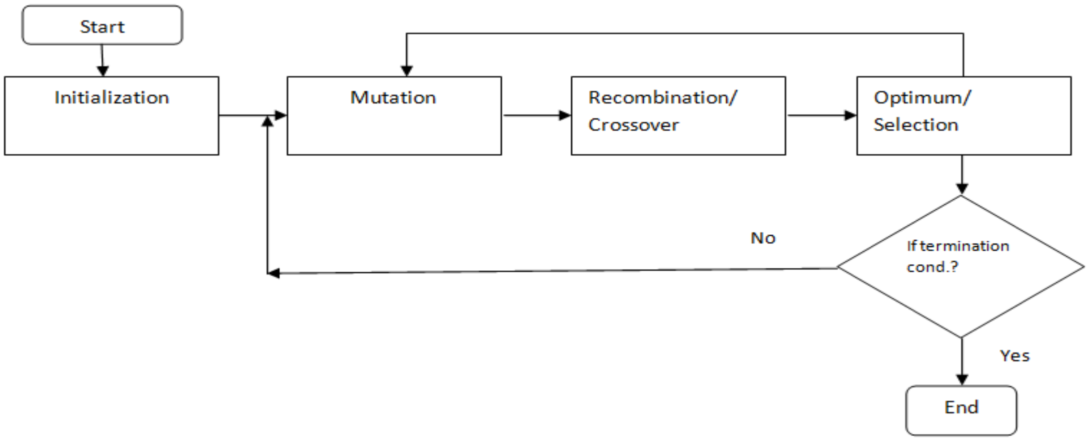


Figure 3.5: The block diagram of traditional DE algorithm

Different Steps of Differential Evolution

Step 1: Initialization

Suppose we want to optimise a function with D real parameters. At first, we have to select the size of the population P_S. The parameter vectors have the form: $X_{i,G} = [X1_{i,G}, X2_{i,G}, ...XD_{i,G}]$, i = 1, 2, . . . , *N*, where *G* is the generation number. Define upper and lower bounds for each parameter: $Xj^L \leq Xj \leq Xj^U$. Now randomly select the initial parameter values uniformly on the intervals $[Xj^L, Xj^U]$. All parameter vectors undergo the mutation, recombination and selection operators.

Step 2: Mutation

Mutation operator expands the search space. For a given parameter vector $X_{i,G}$, randomly select three vectors $X_{r1,G}, X_{r2,G}\, and\, X_{r3,G}$ such that the indices $i, r1, r2\, and\, r3$ are distinct. Add the weighted difference of two of the vectors to the third $V_{i,G+1} = X_{r1,G} + F(X_{r2,G} - X_{r3,G})$. The mutation factor F is a constant from [0, 2]. $V_{i,G}$ is called the donor vector.

The following are the 10 different strategies of DE proposed by Storn and Price [277] for DE, based on the individual being perturbed, the number of individuals used in the mutation process and the type of crossover:

i) DE/best/1/exp

ii) DE/rand/1/exp

iii) DE/rand-to-best/1/exp

iv) DE/best/2/exp

v) DE/rand/2/exp

vi) DE/best/1/bin

vii) DE/rand/1/bin

viii) DE/rand-to-best/1/bin

ix) DE/best/2/bin

x) DE/rand/2/bin

Each strategy generates trial vectors by adding the weighted difference between other randomly selected members of the population. The general convention used above is ”DE/a/b/c“, where DE stands for differential evolution, ‘a’ represents a string denoting the vector to be perturbed, ‘b’ is the number of difference vectors considered for perturbation of ‘a’, and ‘c’ stands for the type of crossover being used (exp: exponential; bin: binomial). For example, the strategy implemented here “DE/rand/1/bin”, means that the target vector is randomly selected, and only one difference vector is used. The ‘bin’ acronym indicates that the recombination is controlled by a binomial decision rule.

In this thesis, we have introduced two new strategies and those are DE/CURRENT_to_BEST/1/power and DE/CURRENT_to_BEST/2/power respectively. Here DE stands for differential evolution, ‘CURRENT_to_BEST’ represents a string denoting that the perturbed vector is approaching towards the best personal best solution for that particular chromosome, ‘1/2’ is the number of difference vectors considered for perturbation of ‘CURRENT_to_BEST’, and ‘power’ stands for the type of crossover being used (power: power crossover, which has been explained previously).

Step 3: Recombination / Crossover

Recombination incorporates successful solutions from the previous generation. The trial vector $u_{i,G+1}$ is developed from the elements of the target vector, $X_{i,G}$, and the elements of the donor vector, $V_{i,G+1}$. Elements of the donor vector enter the trial vector with probability CR.

Step 4: Selection / Optimum

The target vector $X_{i,G}$ is compared with the trial vector $V_{i,G+1}$ and the one with the lowest function value (in case of a minimization problem) is admitted for the next generation. Mutation, recombination and selection continue until a given stopping criterion is reached.

Algorithm for DE

```
Step-1: Set population size ($P_s$), maximum number of generations ($M_g$),
        crossover/ recombination probability ($CR$), mutation probability ($F$) and
        the bounds of decision variables.
Step-2: Set the iteration number $t = 0$.
Step-3: Initialize the chromosomes of the current population $P(t)$.
Step-4: Compute the fitness function for each chromosome of $P(t)$.
Step-5: Find the chromosome having the best fitness value.
Step-6: Set $t = t + 1$.
Step-7: If the termination condition is satisfied, then go to step-13; otherwise go to
        the next step.
Step-8: Improve the best solution of each chromosome by comparing the solutions
of all chromosomes of $P_{DE}(t)$.
Step-9: Apply the mutation operator to compute the donor vector for different DE
        strategies
Step-10: Apply the recombination/crossover operator to obtain the trial vector
from the target vector and the donor vector.
Step-12: Find the better of the best chromosome of $P(t) \& P(t-1))$ and store it;
        then go to Step-6.
Step-13: Print the best chromosome and its fitness value.
Step-14: End.
```

3.3.4 Ant Colony Optimization (ACO)

In Ant Colony Optimization, a number of artificial ants build solutions to the considered optimization problem and exchange information on the quality of these solutions via a communication media, pheromone trail, which is reminiscent of the one adopted by real ants. The original ACO algorithm, known as Ant System, was proposed in the early nineties [279].

Each ant constructs a solution by repeatedly applying a state transition rule and the solution is improved by a local search algorithm. Then the ant modifies the amount of pheromone on the visited edges by applying a local pheromone updating rule. Once all ants have done their operations, the amount of pheromone is modified by applying a global updating rule.

$$\tau_{ij} = \begin{cases} (1-\rho).\tau_{ij} + \rho.\Delta\tau_{ij}, \text{if(i,j)} \in \text{best solution} \\ \quad\quad \tau_{ij} \quad\quad\quad \text{otherwise} \end{cases} \tag{3.42}$$

The local pheromone updating rule is shown in

$$\tau_{ij} = \{\tau_{ij}.(1-\varphi) + \varphi.\tau_0 \tag{3.43}$$

The steps of the proposed meta-heuristic ACO algorithm [280], has been given in figure 3.6.

Step 1: Initialize ANT (generally tallied with population) solution and generate ANT/Population size P_S. Define Attractiveness (τ) and Visibility Function (η). Set the bounds of decision variables.

Step 2: Set the generation/iteration number $t = 0$.

Step 3: Initialize randomly the ANT population of solutions $P_{ANT}(t) = \{x_i(t); i=1,...,NP\}$.

Step 4: Repeat the following until termination criterion is met.

4.1: Increment generation by: $t= t+1$.

4.2: Compute the next ANT solution according to the Attractiveness (τ) and Visibility Function (η). Here, the Visibility function is tallied with the fitness function [$f(x_i)$] for each variable x_i of $P_{ANT}(t)$ and attractiveness is based upon the local Pheromone updating rule .

4.3: Find the global best ANT solution (P_g^{ANT}) having the best fitness/Visibility value depending upon the global Pheromone updating rule and local updating rule and choose the best one.

Step 5: Verify the termination criterion. If the termination conditions are not met, go back to Step 4, elsewhere go to Step 6.

Step 6: Print the value of fitness/Attractiveness value of the global best ANT solution.

Step 7: End.

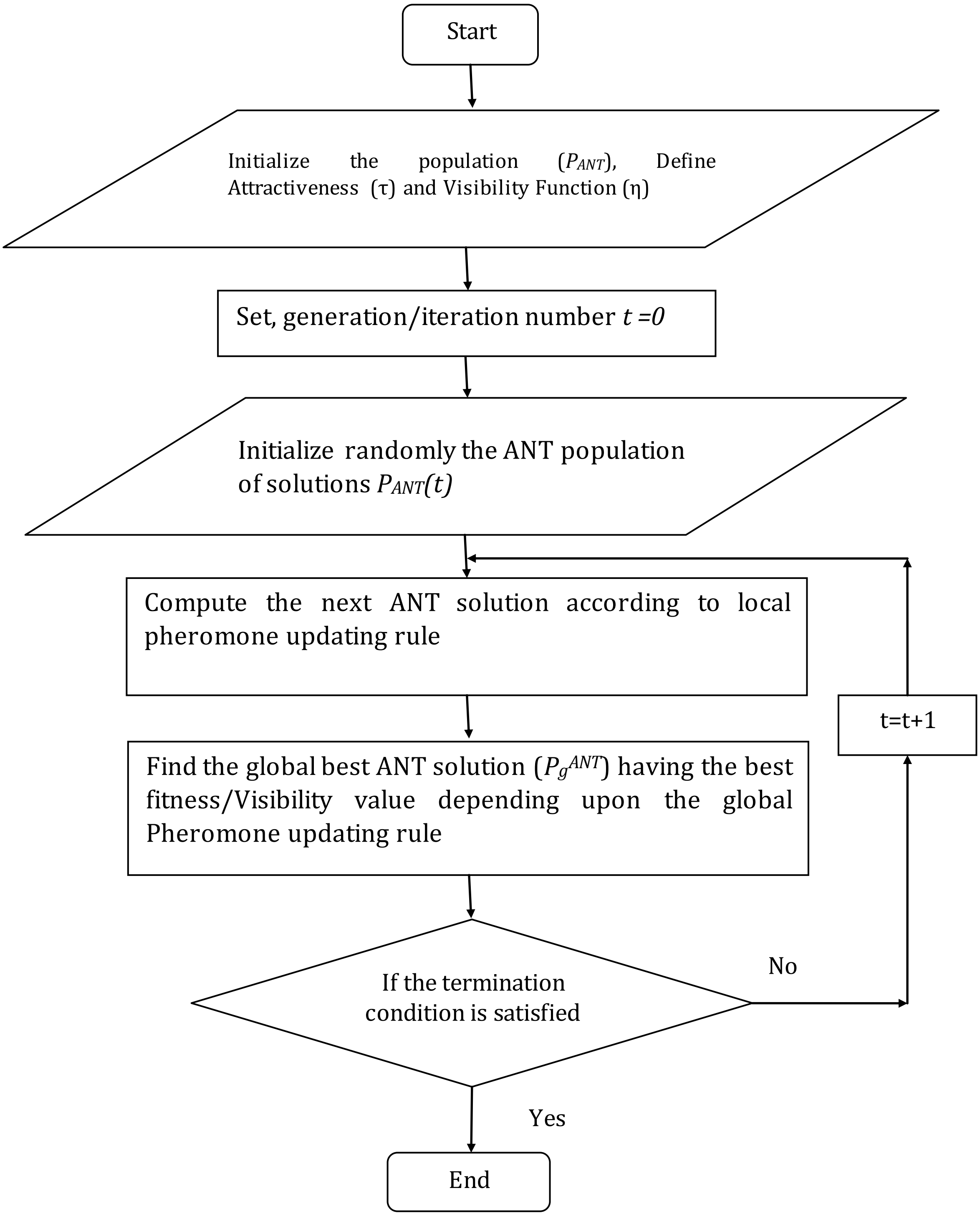

Start
Initialize the population (P_{ANT}), Define Attractiveness (τ) and Visibility Function (η)
Set, generation/iteration number $t = 0$
Initialize randomly the ANT population of solutions $P_{ANT}(t)$
Compute the next ANT solution according to local pheromone updating rule
Find the global best ANT solution (P_g^{ANT}) having the best fitness/Visibility value depending upon the global Pheromone updating rule
If the termination condition is satisfied
No
t=t+1
Yes
End

Figure 3.6: The flowchart of the Ant Colony Optimization (ACO) Algorithm

3.3.5 The GA-PSO Hybrid Algorithm

For solving optimization problems (single objective as well as multiple objectives) we have proposed a novel hybrid algorithm combining two well-known meta-heuristic methods, viz. Genetic Algorithms and Particle Swarm Optimization. We call this hybrid algorithm as GA-PSO.

Motivation of the proposed GA-PSO hybrid algorithm

One of the objectives of the research work is to develop an efficient hybrid approach based on genetic algorithm and particle swarm optimization for solving mixed-integer nonlinear reliability optimization problems. As the development of efficient hybrid algorithm is a demanding trends so we have proposed an alternative hybrid approach of GA–PSO applying GA for 50% chromosomes / particles and PSO for the rest. In each chromosome of GA, the first 50% genes are corresponding to integer variables and the remaining 50% genes corresponding to floating point variables. In this work, we have also proposed a new crossover operator. We have called this operator as power crossover. In this operation, the reduced value of gene corresponding to each chromosome is the intermediate value between the corresponding genes. So this operation produces feasible offspring when two feasible parent chromosomes are crossed. For the improvement of the first 50% genes of each chromosomes, intermediate crossover and one-neighborhood mutation have been applied whereas for the remaining genes, power crossover and uniform mutation have been used. In each iteration/generation of PSO, the particle best position is considered by comparing the population of GA. On the other hand, the global best particle of PSO is obtained by comparing both populations.

In this algorithm, we have applied GA for 50% of the chromosomes and PSO for the rest. The flowchart of GA-PSO algorithm has been shown in figure 3.8. The different steps of the proposed algorithm are as follows:

The GA-PSO Algorithm

```
Step-1: Set population size (2*$P_s$), maximum number of generations ($M_g$),
        crossover probability ($P_c$), mutation probability ($P_m$) and decision variables bounds.
Step-2: Set $t=0$.[ the generation/iteration number]
Step-3: Initialize the chromosomes/particles of the population $P(t)$.
Step-4: Compute the fitness function for each chromosome of $P(t)$.
Step-5: Find the global best chromosome/particle ($P_g$) having the best fitness value.
Step-6: Divide the chromosomes/particles into two groups, viz $P_{GA}(t)$ and $P_{PSO}(t)$
with equal population size.
Step-7: Repeat the following until the termination criterion is satisfied:
        (i)   Increase the value of t by unity.
        (ii)  Apply GA for population $P_{GA}(t)$.
              a. Apply the crossover & mutation operators on $P_{GA}(t)$ to produce a new population $P_{GA}(t)$.
              b. Find the best chromosome ($P'_g$) from the current population $P_{GA}(t)$.
              c. Compare $P'_g$ with the earlier best chromosome $P_g$ and store the better one in $P_g$.
              d. Set $t=t+1$.
              e. Select the population $P_{GA}(t)$ from the population $P_{GA}(t-1)$ of
                 $(t-1)-\mathrm{th}$ generation using tournament selection.
        (iii) Apply PSO-Co for $P_{PSO}(t)$.
              a. Improve the best position of each particle by comparing the position of all chromosomes of $P_{GA}(t)$.
              b. Compute the velocity of each particle.
              c. Obtain the new position of each particle.
              d. Improve the position of each particle and also find the global best particle ($P_g$).
Step-8: Print the position and fitness of global best particle.
Step-9: End.
```

The block diagram of this hybrid algorithm is shown in Figure 3.7.

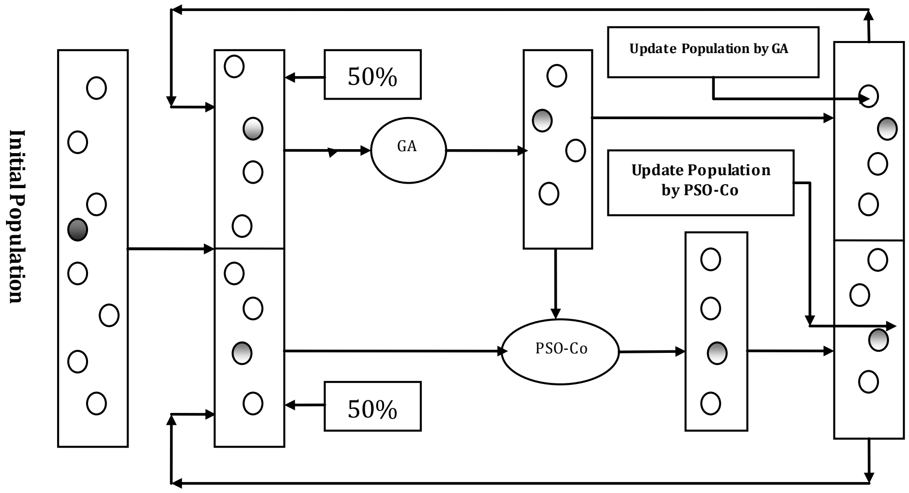


Figure 3.7: The block diagram of the GA-PSO algorithm

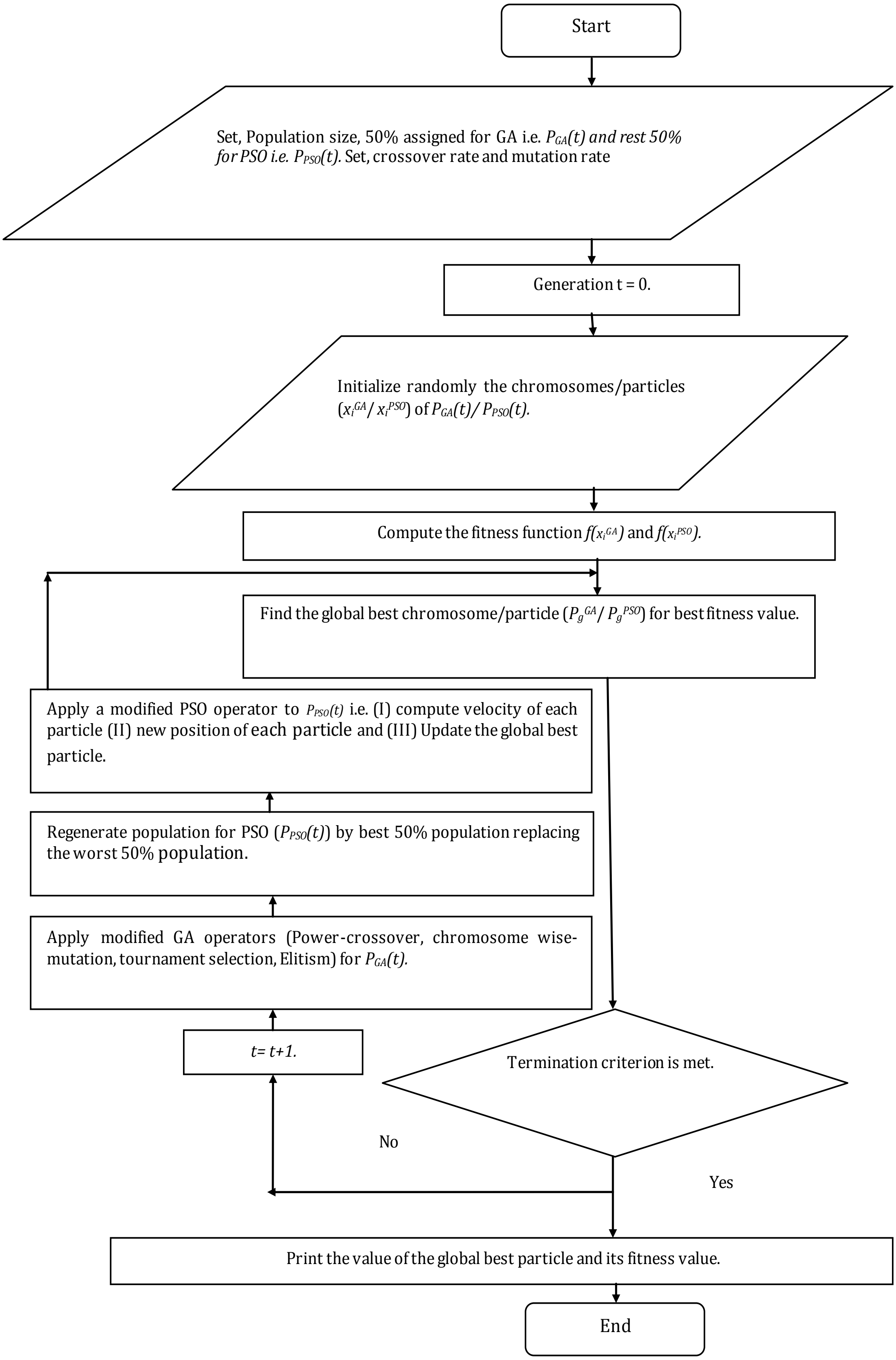

Start
Set, Population size, 50% assigned for GA i.e. $P_{GA}(t)$ and rest 50% for PSO i.e. $P_{PSO}(t)$. Set, crossover rate and mutation rate
Generation t = 0.
Initialize randomly the chromosomes/particles (x_i^{GA}/ x_i^{PSO}) of $P_{GA}(t)$/ $P_{PSO}(t)$.
Compute the fitness function $f(x_i^{GA})$ and $f(x_i^{PSO})$.
Find the global best chromosome/particle (P_g^{GA}/ P_g^{PSO}) for best fitness value.
Apply a modified PSO operator to $P_{PSO}(t)$ i.e. (I) compute velocity of each particle (II) new position of each particle and (III) Update the global best particle.
Regenerate population for PSO ($P_{PSO}(t)$) by best 50% population replacing the worst 50% population.
Apply modified GA operators (Power-crossover, chromosome wise-mutation, tournament selection, Elitism) for $P_{GA}(t)$.
$t= t+1$.
Termination criterion is met.
No
Yes
Print the value of the global best particle and its fitness value.
End

Figure 3.8: Flowchart of GA-PSO Algorithm

The hybrid GA-ACO Algorithm

A hybrid algorithm has been designed in which the GA algorithm (with modified GA operators) is applied first and then the ACO algorithm is applied. This process continues till termination condition is met. Figure 3.9 depicts the process of generating population for GA and ACO. The whole population at any time is used by GA. This population is cloned and the better half is used by ACO as depicted in Figure 3.9. After applying modified GA operators, a modified population is produced. The population for ACO is generated by replacing the 50% best population generated by the GA with the 50% best population retrieved from the cloned population. The flowchart of proposed GA-ACO algorithm has been shown in figure 3.10.

Motivation of the proposed GA-ACO hybrid algorithm

The major learning in this exercise is that hybrid algorithm combining GA and ACO is superior to individual GA and ACO for solving the problem at hand. It is also interesting to note that the operators used for the hybrid algorithm and the combination methodology has a positive impact on the performance of the hybrid algorithm.

This is demonstrated by the fact that our combination performs better than the hybrid algorithm proposed in Lee et al.(2008)[335]. The rationale behind choosing GA and ACO as candidates for hybridization is as follows. GA is selected as a global search algorithm because of its robustness ; GA is further improved upon by using a power-crossover operator (Sahoo et al. 2014)[204] and a non-uniform mutation operator (Zhao et al. 2007)[331]. We are not using the elitism property of GA to induce greediness. This is because we are hybridizing GA with ACO, a known local search technique, which is also considered as "greedy".

In ACO, ants probabilistically find optimal solutions by refining their trajectory in the local search space.

As the refinement process has the property named "positive feedback for rapid detection of good solutions", ACO is fast and can adapt itself to changing situations (Pastorino 2007)[332]. So, we have chosen ACO as the local search technique.

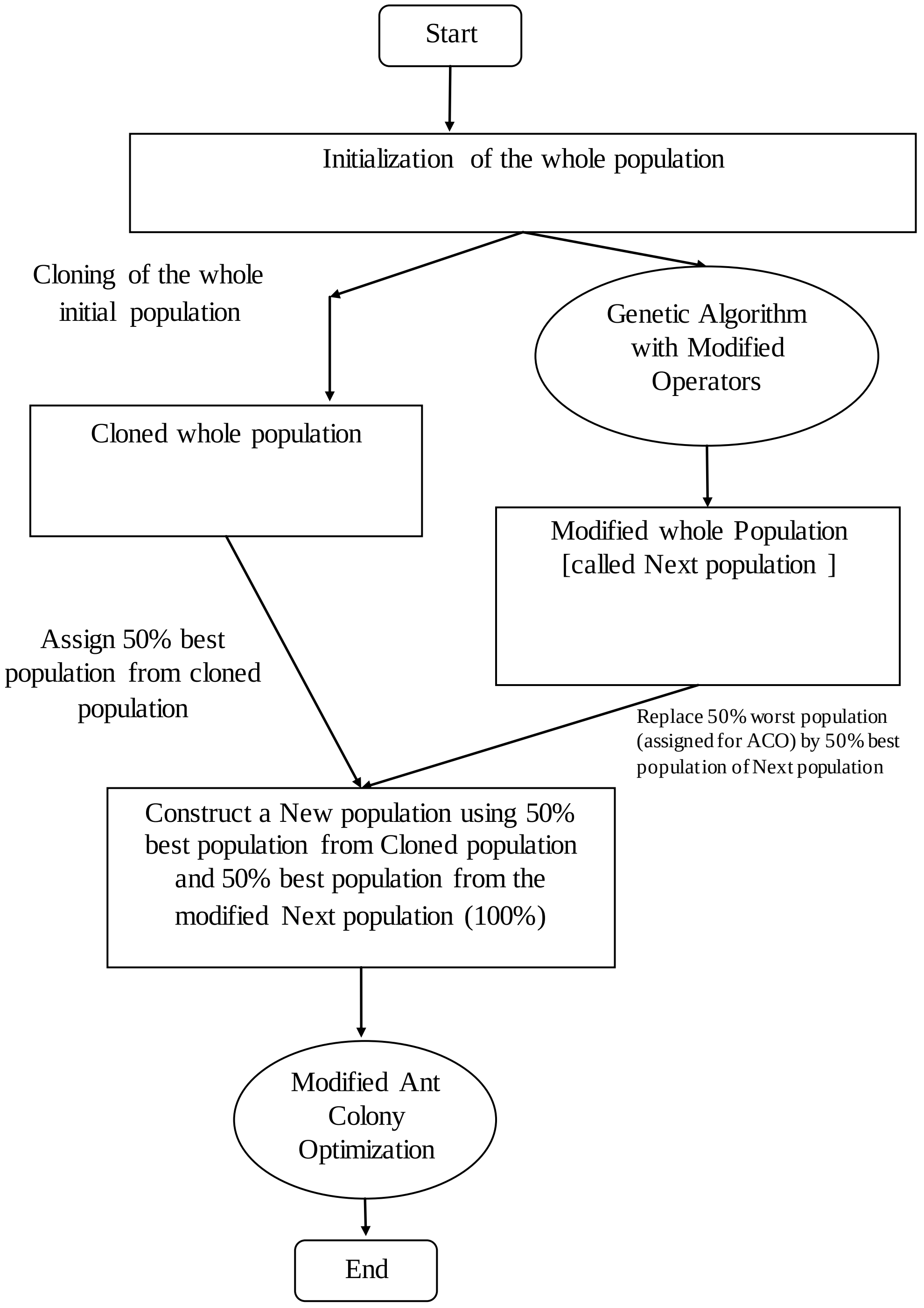

Start
Initialization of the whole population
Cloning of the whole initial population
Genetic Algorithm with Modified Operators
Cloned whole population
Modified whole Population [called Next population]
Assign 50% best population from cloned population
Replace 50% worst population (assigned for ACO) by 50% best population of Next population
Construct a New population using 50% best population from Cloned population and 50% best population from the modified Next population (100%)
Modified Ant Colony Optimization
End

Figure 3.9: The diagram showing the process of generation of population for Genetic Algorithm (GA) and Ant Colony Optimization (ACO) algorithms.

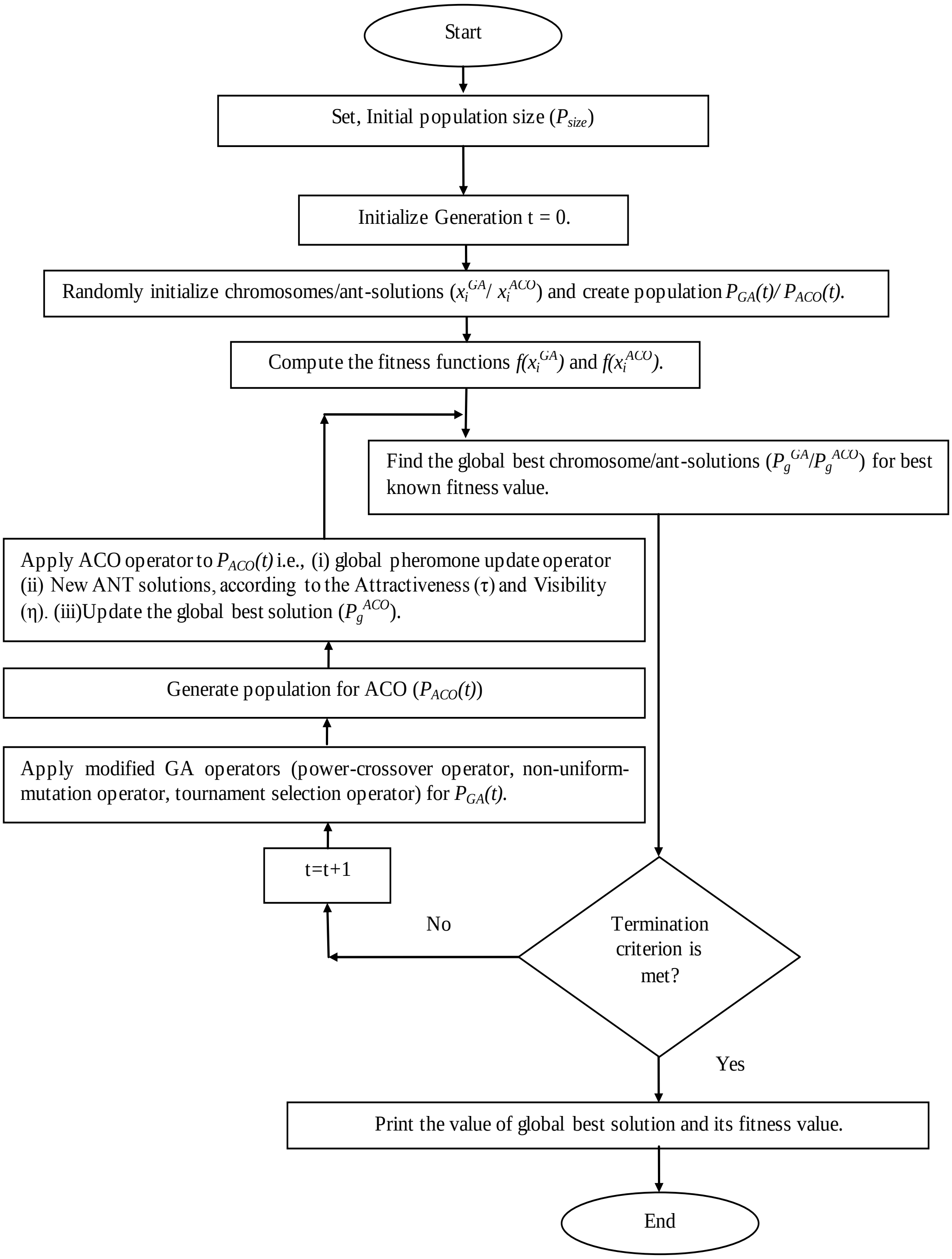

Start
Set, Initial population size (P_{size})
Initialize Generation t = 0.
Randomly initialize chromosomes/ant-solutions (x_i^{GA}/ x_i^{ACO}) and create population $P_{GA}(t)$/ $P_{ACO}(t)$.
Compute the fitness functions $f(x_i^{GA})$ and $f(x_i^{ACO})$.
Find the global best chromosome/ant-solutions (P_g^{GA}/P_g^{ACO}) for best known fitness value.
Apply ACO operator to $P_{ACO}(t)$ i.e., (i) global pheromone update operator (ii) New ANT solutions, according to the Attractiveness (τ) and Visibility (η). (iii)Update the global best solution (P_g^{ACO}).
Generate population for ACO ($P_{ACO}(t)$)
Apply modified GA operators (power-crossover operator, non-uniform-mutation operator, tournament selection operator) for $P_{GA}(t)$.
t=t+1
No
Termination criterion is met?
Yes
Print the value of global best solution and its fitness value.
End

Figure 3.10: Flowchart of the proposed GA-ACO hybrid algorithm

The GA-ACO Algorithm

Step-1: Set population size (2* P_s), maximum number of generations (M_g), crossover probability (P_c), mutation probability (P_m) and decision variables bounds.

Step-2: Set $t=0$.[the generation/iteration number]

Step-3: Initialize chromosomes/ant-solutions (x_i^{GA}/x_i^{ACO}) and create population $P_{GA}(t)/P_{ACO}(t)$.

Step-4: Compute the fitness functions $f(x_i^{GA})$ and $f(x_i^{ACO})$.

Step-5: Find the global best chromosome/ant-solutions (P_g^{GA}/P_g^{ACO}) for best known fitness value.

Step-6: Divide the chromosomes/particles into two groups, viz $P_{GA}(t)$ and $P_{ACO}(t)$ with equal population size.

Step-7: Repeat the following until the termination criterion is satisfied:

(i) Increase the value of t by unity.

(ii) Apply GA for population $P_{GA}(t)$.

a. Apply the crossover & mutation operators on $P_{GA}(t)$ to produce a new population $P_{GA}(t)$.

b. Find the best chromosome (P_g') from the current population $P_{GA}(t)$.

c. Compare P_g' with the earlier best chromosome P_g and store the better one in P_g.

d. Set $t=t+1$.

e. Select the population $P_{GA}(t)$ from the population $P_{GA}(t-1)$ of $(t-1)-\text{th}$ generation using tournament selection.

(iii) Apply ACO for $P_{ACO}(t)$.

a. Improve the Global pheromone update operator.

b. Compute New ANT solutions, according to the Attractiveness (τ) and Visibility (η).

c. Update the global best solution (P_g^{ACO}).

Step-8: Print the value of global best solution and its fitness value.

Step-9: End.

3.4 Concluding Remarks

In this chapter, we have discussed 4 different evolutionary algorithms (Genetic Algorithm, Particle Swarm Optimization, Differential Evolution and Ant Colony Optimization) and we have discussed two different hybrid algorithms (newly developed) i.e., hybrid GA-PSO algorithm and hybrid GA-ACO algorithm. There are some modified operators and unique strategies which make those algorithms different from the traditional evolutionary algorithms. In the hybrid GA-PSO algorithm, a GA algorithm has been applied for 50% of the chromosomes and PSO for the rest; For the improvement of the first 50% genes of each chromosome, which denote the integer part of the problem, intermediate crossover and one-neighbourhood mutation have been applied; on the other hand, for the remaining 50% genes, which denote the real part of the problem, power crossover and uniform mutation have been used. In each iteration/generation of PSO, the particle best position is considered by comparing with the GA population. .In the hybrid GA-ACO algorithm the process of generation of population for Genetic Algorithm (GA) and Ant Colony Optimization (ACO) algorithms has been described using figure 3.9.

CHAPTER 4

GA-PSO Algorithm for mixed-integer nonlinear programming problem in reliability optimization

4.1 Introduction

Due to the development of modern technology, the design of a system and system reliability are more important in industries, especially in complex manufacturing systems. The most fundamental goal of the reliability optimization is how to improve the system reliability subject to the resource/budget constraints. The basic objective of reliability redundancy allocation problem is to find the number of redundant components and also the reliability levels of each component that either maximize the system reliability or minimize the system cost under several constraints. Reliability- Redundancy allocation problem (RRAP) is basically a nonlinear mixed-integer programming problem. Most of these problems cannot be solved by direct/indirect or mixed search methods due to discrete search space. According to Chern [281], reliability redundancy allocation problem with multiple constraints is quite often hard to find feasible solutions and this type of problem fall in the category of NP-hard. There are basically four types of system configurations viz. series, parallel, series-parallel and complex/complicated. In reliability redundancy allocation problem it is seen that some of the decision variables are integer variables and others are continuous variables. So, finding of optimal solutions to such combinatorial optimization problems is a formidable task to the decision makers'. Earlier, several deterministic methods like heuristic methods have been discussed by Nakagawa and Nakashima, Kuo et al., Kim and Yum, Aggarwal and Gupta, Kulturel-Konak et al. etc. [282-286]. The reduced gradient method [287], branch and bound method [287-290], integer programming [291], dynamic programming [292,293] and other well-developed mathematical programming techniques were used to solve such redundancy allocation problem. However, these methods have both advantages and disadvantages. Dynamic programming is not useful for reliability optimization of a general system as it can be used only for few particular structures of the objective function and constraints that are decomposable. In branch and bound method, the effectiveness depends on sharpness of the bound and required memory increases exponentially with the problem size. As a result, with the development of heuristic techniques like genetic algorithm, particle swarm optimization researchers focus on these methods as that provides a reasonable solution to a complex combinatorial optimization problem within a reasonable time complexity. To improve computational efficiency researchers have used hybrid algorithms to meet their individual goals. In this connection, genetic algorithm, ant colony optimization, simulated annealing and particle swarm optimization have been successfully applied for solving reliability optimization problems. To hybridized, GA has been combined with other heuristic algorithms to attain efficiency from the computational point of view.

Applications of Genetic algorithms in reliability optimization problems have been reported in the works of Dengiz et al. [294], Tavakkoli-Moghaddama et al. [295], Ye et al. [296], Gupta et al. [297], Bhunia et al. [298], Sahoo et al. [299,300],Mahato et al. [301] and Sahoo et al. [302,303]. Using genetic algorithm, mixed-integer nonlinear reliability problems have been solved by Hsieh et al. [304] considering series, parallel, series-parallel and complex systems configurations. The application of simulated annealing in optimal reliability design has been reported in the work of Kuo et al. [305]. Kim and Bae [306] have solved RRAP using simulated annealing and showed that their solution is better in compare to the works of Hitika et al. [307], Hsieh et al. [308] and Yokota et al. [309]. Similar work has been found in the existing literature considering series and complex (bridge) systems using particle swarm optimization (PSO) algorithm [310].

4.2 Formulation of Reliability-Redundancy Optimization Problems

The objective of the Reliability Redundancy Allocation problem is to enhance the reliability of the system under certain resource constraints based on the cost, weight and volume of the system. This objective can be fulfilled by adding an appropriate number of more reliable redundant components. Hence, the mathematical formulation of the Reliability Redundancy Allocation problem is as follows:

$$\text{Maximize}\, R_s = f(x_1, x_2, ..., x_n; r_1, r_2, ..., r_n) \tag{4.1}$$

subject to $g_i(x_1, x_2, ..., x_n; r_1, r_2, ..., r_n) \le b_i$

where $x_i \in Z^+$, $0 \le r_i \le 1$ and $i = 1, 2, ..., m$

where R_s is the system reliability and $g_i(r, x)$ is the *i*-th constraint function, which are associated with system weight, volume and cost. The aim of this problem is to determine the number of components and the components' reliability in each system so as to maximize the overall system reliability. The problem belongs to the category of constrained nonlinear mixed-integer programming problems.

Problem 4.1: Series System

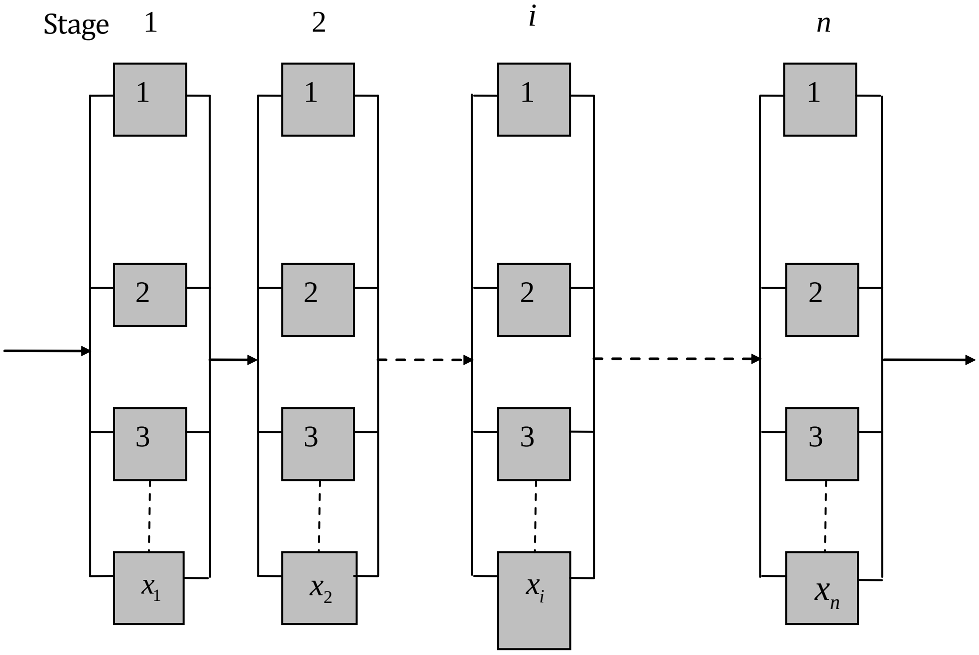


Figure 4.1: The structure of series system

Figure 4.1 shows the series system. The corresponding optimization problem of a series system [204] is as follows:

$$\text{Maximize } f(r,x) = \prod_{i=1}^{m} R_i(r_i, x_i) \qquad (4.2)$$

subject to

$$g_1(r,x) = \sum_{i=1}^{m} v_i x_i^2 \le V$$

$$g_2(r,x) = \sum_{i=1}^{m} \alpha_i \left(\frac{-1000}{\ln r_i}\right)^{\beta_i} \left(x_i + \exp\left(\frac{x_i}{4}\right)\right) \le C$$

$$g_3(r,x) = \sum_{i=1}^{m} w_i x_i \exp(\frac{x_i}{4}) \le W$$

where $R_i(r_i, x_i) = 1 - (1 - r_i)^{x_i}$

Problem 4.2: Series-parallel system

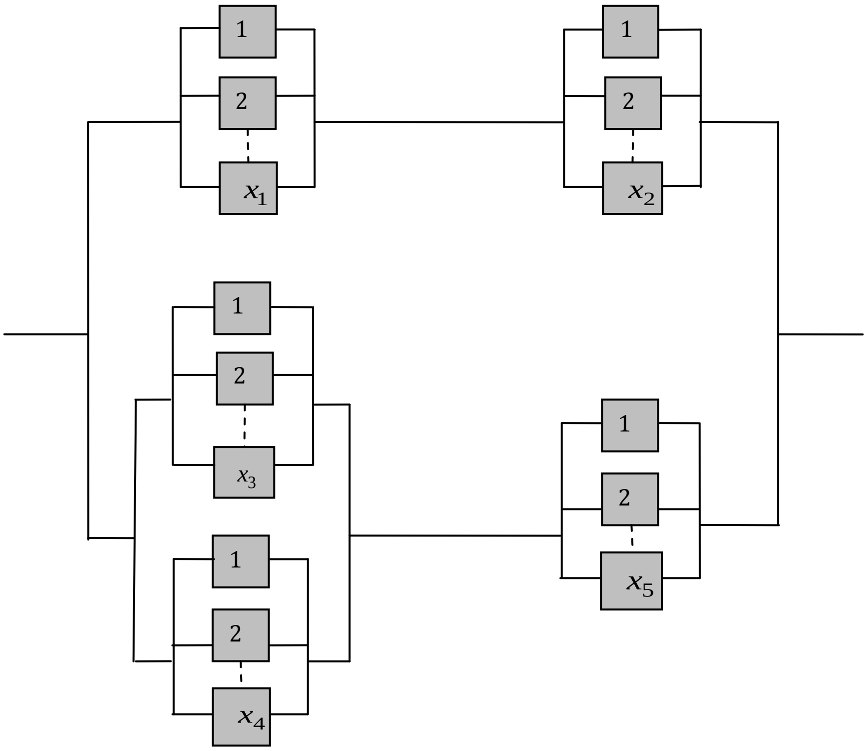


Figure 4.2: The structure of series-parallel system

Figure 4.2 shows the series-parallel system. The corresponding optimization problem of series-parallel systems [306,311]) is as follows:

$$\text{Maximize } f(r,x) = 1-(1-R_1R_2)(1-(1-R_3)(1-R_4)R_5) \qquad (4.3)$$

subject to

$$g_1(r,x) = \sum_{i=1}^{m} v_i x_i^2 \le V$$

$$g_2(r,x) = \sum_{i=1}^{m} \alpha_i \left(\frac{-1000}{\ln r_i}\right)^{\beta_i} \left(x_i + \exp\left(\frac{x_i}{4}\right)\right) \le C$$

$$g_3(r,x) = \sum_{i=1}^{m} w_i x_i \exp\left(\frac{x_i}{4}\right) \le W$$

and $R_i(r_i, x_i) = 1-(1-r_i)^{x_i}$

where $x_i \in Z^+$, where Z^+ is the discrete space of integers, and $0 \le r_i \le 1$, $r_i \in \Re$ where $\Re$ is set of real numbers, $0 \le i \le m$.

Problem 4.3: Complex (bridge) system

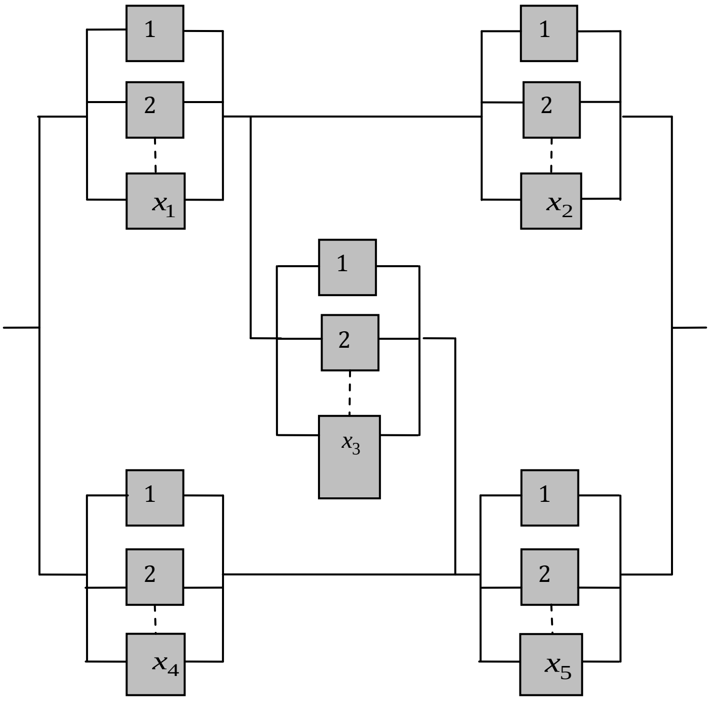


Figure 4.3: The structure of complex (bridge) system

Figure 4.3 shows the complex (bridge) system. The corresponding optimization problem of complex (bridge) systems is as follows:

$$\text{Maximize } f(r,x) = R_1R_2 + R_3R_4 + R_1R_4R_5 - R_1R_2R_3R_4 - R_1R_2R_3R_5 - R_1R_3R_4R_5 - R_1R_2R_4R_5 + 2R_1R_2R_3R_4R_5$$

subject to (4.4)

$$g_1(r,x) = \sum_{i=1}^{m} v_i x_i^2 \leq V$$

$$g_2(r,x) = \sum_{i=1}^{m} \alpha_i \left(\frac{-1000}{\ln r_i}\right)^{\beta_i} \left(x_i + \exp\left(\frac{x_i}{4}\right)\right) \leq C$$

$$g_3(r,x) = \sum_{i=1}^{m} w_i x_i \exp\left(\frac{x_i}{4}\right) \leq W$$

and $R_i(r_i, x_i) = 1 - (1 - r_i)^{x_i}$

where $x_i \in Z^+$, where Z^+ is the discrete space of integers, and $0 \leq r_i \leq 1$, $r_i \in \Re$, where $\Re$ is the set of real numbers, $0 \leq i \leq m$. The parameters α_i and β_i are physical features of system components.

4.3 Solution Procedures

To solve the problem mentioned in this chapter we have used hybrid GA-PSO algorithm. This hybrid algorithm is discussed in detail in the chapter no 3.

4.4 Numerical Solutions, Results and Discussion

4.4.1 The constraint handling technique for constrained mixed-integer nonlinear problems

In the application of the evolutionary algorithm for the given constrained mixed-integer nonlinear optimization problem, an important question arises: how does the algorithm handle the constraints relating to the optimization problem? During the last few decades, several methods have been proposed to handle the constraints for solving constrained optimization problems with the help of evolutionary algorithms. Among these methods, the penalty function method is very popular. In this method, the constrained optimization problem is converted into an unconstrained one in which the reduced objective function involves the original objective function and a penalty for violating the constraints. Recently, Gupta et al. [298] proposed a penalty function approach to handle the constraints. In this approach, in order to convert the constrained optimization problem into an unconstrained one, a large negative value (say, $-M$) is blindly assigned to the objective function for the infeasible solution (for a maximization problem). In this case, if the constrained optimization problem is

$$\text{Maximize}\, R_s = f(x_1, x_2, ..., x_n; r_1, r_2, ..., r_n) \tag{4.5}$$

subject to $g_i(x_1, x_2, ..., x_n; r_1, r_2, ..., r_n) \le b_i$, $i = 1, 2, ... m$

then the reduced unconstrained optimization problem is as follows:

$$\text{Maximize } \hat{f}(x, r) = f(x, r) + \theta(x, r) \tag{4.6}$$

where $\theta(x,r) = \begin{cases} 0 & \text{if } (x,r) \in S \\ -M & \text{if } (x,r) \notin S \end{cases}$

and $S = \{(x,r) : g_i(x,r) - b_i \le 0, i = 1, 2, ..., m\}$ the feasible space for the optimization problem.

For a minimization problem, it is to be noted that instead of $-M$, $+M$ is considered. In this work, we have used the value of M as 99999.

4.4.2 Numerical Examples

In this section, we have considered the Reliability Redundancy Allocation problem in three different systems, viz. series (see Figure 4.1), series-parallel (see Figure 4.2) and complex (bridge) system (Figure 4.3) for numerical experiments. All the values of the parameters related to problems 4.1-4.3 are given in Tables 4.1-4.3: The proposed method has been coded in the C programming language. The computational work has been done on a PC with a 2.10 GHz Intel Core 2 Duo processor and Linux environment. For each example, 50 independent runs have been performed to calculate the best found system reliability which is nothing but the optimal values of system reliability. Also we have computed the maximum (best) and minimum (worst) values of system reliability. These results have been shown in Table 4.4. Also, the statistical measures like mean and standard deviation of system reliability, as well as computational time, have been obtained for comparison with the existing results reported in the literature (cf. Tables 4.5-4.8). In all the cases, the overall system reliability obtained by our proposed method is the best. However, the slack of the first constraint is the lowest whereas the same for third constraint is largest. Also, the slack for the second constraint is quite the same with the earlier reported works. It may also be noted that the average CPU time and standard deviation of CPU time (in seconds) over 50 runs required for implementing the proposed GA-PSO approach is very less, which is presented in Table 9. In this computation, the values of parameters like population size, maximum number of generation, crossover probability rate and mutation probability have been taken as 100, 200, 0.10 and 0.90 respectively.

Table 4.1: Input parameters for series system

Stage	$10^5\alpha$	β_i	v_i	w_i	V	C	W
1	1.0	1.5	1	6	250	400	500
2	2.3	1.5	2	6			
3	0.3	1.5	3	8			
4	2.3	1.5	2	7			

Table 4.2: Input parameters for series-parallel system

Stage	$10^5\alpha$	β_i	v_i	w_i	V	C	W
1	2.500	1.5	2	3.5	180	175	100
2	1.450	1.5	4	4			
3	0.541	1.5	5	4			
4	0.541	1.5	8	3.5			
5	2.100	1.5	4	4.5			

Table 4.3: Input parameters for complex system

Stage	$10^5\alpha$	β_i	v_i	w_i	V	C	W
1	2.330	1.5	1	7	110	175	200
2	1.450	1.5	2	8			
3	0.541	1.5	3	8			
4	8.050	1.5	4	6			
5	1.950	1.5	2	9			

Table 4.4: Best results for series, series parallel and complex systems over 50 runs

Parameter	Series	Series-parallel	Complex
$f(r,x)$	0.99998728	0.99999988	0.99999952
x_1	5	4	4
x_2	6	3	3
x_3	5	2	3
x_4	6	3	3
x_5	-	2	1
r_1	0.915112	0.860328	0.858430
r_2	0.873144	0.827071	0.700000
r_3	0.935557	0.872994	0.922386
r_4	0.879665	0.937884	0.700000
r_5	-	0.701148	0.700000

Parameter	Series	Series-parallel	Complex
Slack of first constraint	6	4	11
Slack of second constraint	0	0	0.000248
Slack of third constraint	97.694289	22.444054	48.888109

Table 4.5: Comparison of the best GA-PSO results with existing algorithms (for series system)

Parameters	Genetic algorithm [309] (Yokota et al.)	Particle swarm approach [310] (Coelho)	Hybrid GA-PSO approach [311] (Sheikhalishahi et al.)	Our proposed approach
$f(r,x)$	0.99994500	0.99995300	0.99995467	0.99998728
x_1	5	5	5	5
x_2	5	6	5	6
x_3	5	4	4	5
x_4	5	5	6	6
r_1	0.895644	0.902231	0.901628	0.915112
r_2	0.885878	0.856325	0.888230	0.873144
r_3	0.912184	0.948145	0.948121	0.935557
r_4	0.887785	0.883156	0.849921	0.879665
Slack of first constraint	50	55	55	6
Slack of second constraint	0.938000	0.975465	0.000006	0
Slack of third constraint	28.803700	24.801882	15.363463	97.694289

Table 4.6: Comparison of the best GA-PSO results with existing algorithms (for series-parallel system)

Parameters	Genetic algorithm [306] (Hsieh et al.)	Simulated annealing Algorithms [308] (Kim and Bae)	Hybrid GA-PSO approach [301] (Sheikhalishahi et al.)	Our proposed approach
$f(r,x)$	0.99997418	0.99997631	0.99997665	0.99999988
x_1	2	2	2	4
x_2	2	2	2	3
x_3	2	2	2	2
x_4	2	2	2	3
x_5	4	4	4	2
r_1	0.785452	0.812161	0.819640	0.860328
r_2	0.842998	0.853346	0.845091	0.827071

Parameters	Genetic algorithm [306] (Hsieh et al.)	Simulated annealing Algorithms [308] (Kim and Bae)	Hybrid GA-PSO approach [301] (Sheikhalishahi et al.)	Our proposed approach
r_3	0.885333	0.897597	0.895482	0.872994
r_4	0.917958	0.900710	0.895517	0.937884
r_5	0.870318	0.866316	0.868430	0.701148
Slack of first constraint	40	40	40	4
Slack of second constraint	1.194440	0.007300	0.000001	0
Slack of third constraint	1.609289	1.609289	1.609289	22.444054

Table 4.7: Comparison of the best GA-PSO results with existing algorithms (for complex system)

Parameters	Genetic algorithm [303] (Hsieh et al.)	Simulated annealing Algorithms [304] (Kim and Bae)	Particle swarm approach [310] (Coelho)	Hybrid GA-PSO approach [312] (Sheikhalishahi et al.)	Our proposed approach
$f(r,x)$	0.99988764	0.99987916	0.99988957	0.99988964	0.99999952
x_1	3	3	3	3	4
x_2	3	3	3	3	3
x_3	3	3	2	2	3
x_4	3	3	4	4	3
x_5	1	1	1	1	1
r_1	0.807263	0.814090	0.826678	0.828134	0.858430
r_2	0.868116	0.864614	0.857172	0.857831	0.700000
r_3	0.872862	0.890291	0.914629	0.914192	0.922386
r_4	0.712673	0.701190	0.648918	0.648069	0.700000
r_5	0.751034	0.734731	0.715290	0.704476	0.700000
Slack of first constraint	40	18	5	5	11
Slack of second constraint	0.007300	0.376347	0.000339	0.000000	0.000248
Slack of third constraint	1.609289	4.264770	1.560466	1.560466	48.888109

Table 4.8: Statistical analysis for series, series-parallel and complex systems

	Problems	Maximum(Best)	Minimum(Worst)	Mean	Standard Deviation
Our proposed approach	Series	0.99998728	0.99995793	0.99998048	6.6107×10^{-6}
	Series-parallel	0.99999988	0.99999566	0.99999917	9.9214×10^{-7}
	Complex	0.99999952	0.99998848	0.99999692	2.7907×10^{-6}
Sheikhalishahi et al. [312]	Series	0.99995467	0.99995467	0.99995467	1.0000×10^{-16}
	Series-parallel	0.99997665	0.99997015	0.99997613	4.5330×10^{-12}
	Complex	0.99988964	0.99988935	0.999889623	2.8226×10^{-11}
Coelho [310]	Series	0.99995300	0.99963800	0.99990700	11.0000×10^{-6}
	Complex	0.99988957	0.99987750	0.99988594	6.9000×10^{-7}

Table 4.9: Average and standard deviation of CPU times (in second) over 50 runs

	(Sheikhalishahi et al. [312])		Our proposed approach	
Problems	Average Time(s)	Standard Deviation(s)	Average Time(s)	Standard Deviation(s)
Series	3.14	0.06	0.14	1.0×10^{-4}
Series-parallel	3.36	0.14	0.18	1.0×10^{-4}
Complex	3.32	0.09	0.18	1.0×10^{-4}

Wilcoxon Rank-Sum Test

This statistical test is used to compare two paired samples (populations) and to calculate the difference between each set of pairs and analyses these differences between matched samples. In this chapter we have performed the Wilcoxon Rank Sum statistical test to compare between two sample populations namely population1 and population2 and compared significance of one population over another one.

Wilcoxon Rank-Sum Test for Series System

Here we have referred population1 for population generated by algorithm using the proposed model of Shikhalashi et al. [312] and population2 for our proposed model.

Table 4.10: Ranks of objective-function value of different populations for series system

Objective function value			Ranks of Objective function value		Objective function value			Ranks of Objective function value	
Run	population1	population 2	Population 1	Population 2	Run	Population 1	Population2	Population 1	Population 2
1	0.999953788	0.999982357	44	78	26	0.999925681	0.999978542	9	72
2	0.999941403	0.99998188	24	76.5	27	0.999936649	0.999974728	16	62.5
3	0.999934837	0.999977112	11	68	28	0.99992014	0.999983929	8	85
4	0.999936635	0.99998045	15	74	29	0.999935728	0.999985218	14	93.5
5	0.999943311	0.999976833	26	67	30	0.999935695	0.999965335	12.5	55
6	0.999931403	0.99998188	10	76.5	31	0.999939006	0.999986692	23	98
7	0.99995045	0.99998728	38	99	32	0.999945913	0.999983067	31	81
8	0.999949496	0.999970436	37	57	33	0.99991528	0.999985695	5.5	96
9	0.99995467	0.999985218	47	93.5	34	0.99995515	0.999953747	48	43
10	0.99995913	0.999980592	51	75	35	0.99994714	0.999964237	33.5	53.5
11	0.99991528	0.999973774	5.5	61	36	0.99994741	0.999964237	35.5	53.5
12	0.999953515	0.999974728	42	62.5	37	0.99993774	0.999984741	17.5	89.5
13	0.99994714	0.999968632	33.5	56	38	0.99991403	0.999975204	3.5	64
14	0.99994741	0.999979496	35.5	73	39	0.99990927	0.99997139	1.5	59.5
15	0.99993774	0.999983525	17.5	84	40	0.99993788	0.99997139	19.5	59.5
16	0.99991403	0.999976158	3.5	66	41	0.99995392	0.999983311	45.5	82.5
17	0.99990927	0.999986172	1.5	97	42	0.999945668	0.999978065	27	70.5
18	0.99993788	0.999984419	19.5	88	43	0.999945695	0.999984264	29.5	86.5
19	0.99995392	0.999985218	45.5	93.5	44	0.99993791	0.999977589	21.5	69
20	0.99995168	0.999961853	39	52	45	0.999915759	0.999984892	7	91
21	0.99995695	0.999971033	49	58	46	0.99995204	0.999984264	40.5	86.5
22	0.99993791	0.999985218	21.5	93.5	47	0.999945695	0.999987281	29.5	100
23	0.99995759	0.999975833	50	65	48	0.999945681	0.999982462	28	79
24	0.99995204	0.999984741	40.5	89.5	49	0.999946649	0.999983311	32	82.5
25	0.999935695	0.999982853	12.5	80	50	0.999942014	0.999978065	25	70.5

Table 4.11: p-value calculation of different populations for series system

	Population1	Population2
count	50	50
Rank-sum	1283	3767
α	0.05	0.05
W’	1283	NA
W"	NA	3767
Mean	0.999939938	0.999978587
Variance	$1.85399X10^{-10}$	$5.72487X10^{-11}$
Standard deviation	$1.40259X10^{-05}$	$7.56629X10^{-06}$
p-value	2	0

Conclusion: Since p-value of population2 (i.e., 0) is less than α value (i.e., 0.05) whereas p-value of population2 (i.e., 2) is greater than α value (i.e., 0.05), therefore the population 2 has significance over population1 in the generated sample (population).

Wilcoxon Rank-Sum Test for Series-Parallel System

Here we have referred population 1 for population generated by algorithm using the proposed model of Shikhalashi et al. [312] and population2 for our proposed model.

Table 4.12: Ranks of objective-function value of different populations for series-parallel system

Objective Function value			Ranks of Object Function value		Objective Function value			Ranks of Object Function value	
Run	population 1	population 2	Population1	Populatio2	Run	Population 1	Population2	Population 1	Population 2
1	0.99997474	0.999999474	41	66	26	0.999975681	0.999999777	45.5	88
2	0.99997277	0.999999777	37	88	27	0.99986649	0.999999257	9	56.5
3	0.9999257	0.999999257	16.5	56.5	28	0.99982014	0.999999257	3	56.5
4	0.9999257	0.999999257	16.5	56.5	29	0.999975728	0.999999474	47	66
5	0.9999474	0.999999474	27	66	30	0.9999474	0.999999777	27	88
6	0.9999777	0.999999777	49	88	31	0.99995777	0.999999257	31	56.5
7	0.9999257	0.999999257	16.5	56.5	32	0.9999257	0.999999257	16.5	56.5
8	0.9999257	0.999999257	16.5	56.5	33	0.9999257	0.999999474	16.5	66
9	0.9999474	0.999999474	27	66	34	0.9999474	0.999999777	27	88
10	0.99997077	0.999999777	36	88	35	0.99992777	0.999999257	23	56.5
11	0.9999257	0.999999257	16.5	56.5	36	0.9999257	0.999999257	16.5	56.5
12	0.9999257	0.999999257	16.5	56.5	37	0.9999257	0.999999688	16.5	77
13	0.99997665	0.99999988	48	92	38	0.9999474	0.999999888	27	96.5
14	0.99984741	0.999999888	5	96.5	39	0.99991777	0.999999688	10	77
15	0.999973774	0.999999688	38	77	40	0.9999257	0.999999688	16.5	77
16	0.99981403	0.999999688	2	77	41	0.9999257	0.999999888	16.5	96.5
17	0.99980927	0.999999888	1	96.5	42	0.999965668	0.999999688	33.5	77
18	0.99983788	0.999999688	4	77	43	0.99985695	0.999999688	6.5	77
19	0.999975392	0.999999688	44	77	44	0.999973791	0.999999888	39.5	96.5
20	0.999965668	0.999999888	33.5	96.5	45	0.99985759	0.999999688	8	77
21	0.999985695	0.999999688	50	77	46	0.999975204	0.999999688	42.5	77
22	0.999973791	0.999999688	39.5	77	47	0.99985695	0.999999888	6.5	96.5
23	0.999958576	0.999999888	32	96.5	48	0.999975681	0.999999688	45.5	77
24	0.999975204	0.999999688	42.5	77	49	0.999938665	0.999999474	24	66
25	0.99994857	0.999999474	30	66	50	0.999968201	0.999999777	35	88

Table 4.13: p-value calculation of different populations for series-parallel system

	Population 1	Population 2
count	50	50
Rank-sum	1275	3775
α	0.05	0.05
W’	1275	NA
W"	NA	3775
Mean	0.999931867	0.999999603
Variance	$2.3324X10^{-09}$	$5.30595X10^{-14}$
Standard deviation	$4.95736X10^{-05}$	$2.30346X10^{-07}$
p-value	2	0

Conclusion: Since p-value of population2 (i.e., 0) is less than α value (i.e., 0.05) whereas p-value of population2 (i.e., 2) is greater than α value (i.e., 0.05), therefore the population 2 has significance over population1 in the generated sample (population).

Wilcoxon Rank-Sum Test for Complex System

Here we have referred population1 for population generated by algorithm using the proposed model of Shikhalashi et al. [312] and population2 for our proposed model.

Table 4.14: Ranks of objective-function value of different populations for complex system

Objective Function value			Ranks of Object Function value		Objective Function value			Ranks of Object Function value	
Run	population 1	population 2	Population 1	Population 2	Run	Population 1	Population 2	Population 1	Population 2
1	0.9991618	0.999997616	5.5	90	26	0.99975681	0.999998635	23.5	95
2	0.999004	0.999989993	1	54.5	27	0.99986649	0.999995708	44.5	83
3	0.9994278	0.999996662	7.5	85.5	28	0.99982014	0.999990033	32.5	57
4	0.9991417	0.999968162	3.5	52	29	0.99975728	0.999990975	25	66.5
5	0.9997616	0.999998691	26.5	98	30	0.9991618	0.999990975	5.5	66.5
6	0.99987993	0.99999094	48.5	62	31	0.9998004	0.999998569	28	92
7	0.9996662	0.999998635	12.5	95	32	0.9994278	0.999992913	7.5	75
8	0.99968162	0.999995708	14.5	83	33	0.9991417	0.999994278	3.5	79
9	0.9998691	0.999990033	46.5	57	34	0.9997616	0.999997139	26.5	88
10	0.9998094	0.999990975	30	66.5	35	0.99987993	0.999991618	48.5	72.5
11	0.9998635	0.999990975	42.5	66.5	36	0.9996662	0.99999004	12.5	59.5
12	0.99958515	0.999998569	9	92	37	0.99968162	0.999994278	14.5	79
13	0.99964714	0.999992913	10	75	38	0.9998691	0.999991417	46.5	70.5
14	0.99984741	0.999994278	35	79	39	0.999094	0.999968162	2	52
15	0.99973774	0.999997139	16	88	40	0.9998635	0.999998691	42.5	98
16	0.99981403	0.999991618	31	72.5	41	0.99975392	0.99999094	21.5	62
17	0.99980927	0.99999004	29	59.5	42	0.99965668	0.999998635	11	95
18	0.99983788	0.999994278	34	79	43	0.99985695	0.999995708	37.5	83
19	0.99975392	0.999991417	21.5	70.5	44	0.99973791	0.999990033	17.5	57
20	0.99988964	0.99999952	50	100	45	0.99985759	0.999990975	40.5	66.5
21	0.99985695	0.999989993	37.5	54.5	46	0.99975204	0.999990975	19.5	66.5
22	0.99973791	0.999996662	17.5	85.5	47	0.99985695	0.999998569	37.5	92
23	0.99985759	0.999968162	40.5	52	48	0.99975681	0.999992913	23.5	75
24	0.99975204	0.999998691	19.5	98	49	0.99986649	0.999994278	44.5	79
25	0.99985695	0.99999094	37.5	62	50	0.99982014	0.999997139	32.5	88

Table 4.15: p-value calculation of different populations for complex system

	Population 1	Population 2
count	50	50
Rank-sum	1275	3775
α	0.05	0.05
W’	1275	NA
W"	NA	3775
Mean	0.999694282	0.999992524
Variance	$5.68613X10^{-08}$	$4.90882X10^{-11}$
Standard deviation	0.000186141	$7.0063X10^{-06}$
p-value	2	0

Conclusion: Since p-value of population2 (i.e., 0) is less than α value (i.e., 0.05) whereas p-value of population2 (i.e., 2) is greater than α value (i.e., 0.05), therefore the population 2 has significance over population1 in the generated sample (population).

Graphical representations of the results

Graphical representations of the results have been implemented for reliability optimization of series, series-parallel and complex system using proposed GA-PSO algorithm. Figure 4.4 represents the convergence characteristics of single objective – reliability optimization in series system using proposed GA-PSO algorithm for 200 generations of a population, Figure 4.5 represents the convergence characteristics of single objective – reliability optimization in series system using proposed GA-PSO algorithm for 200 generations of a population and Figure 4.6 represents the convergence characteristics of single objective – reliability optimization in series system using proposed GA-PSO algorithm for 200 generations of a population. The diagrams show that the objective functions quickly converge towards global optima or become close to the minimum possible values starting from some initial value. For showing the convergence 2-Dimensional coordinate system has been used. X-axis denotes reliability; Y-axis denotes the number of generation in a population.

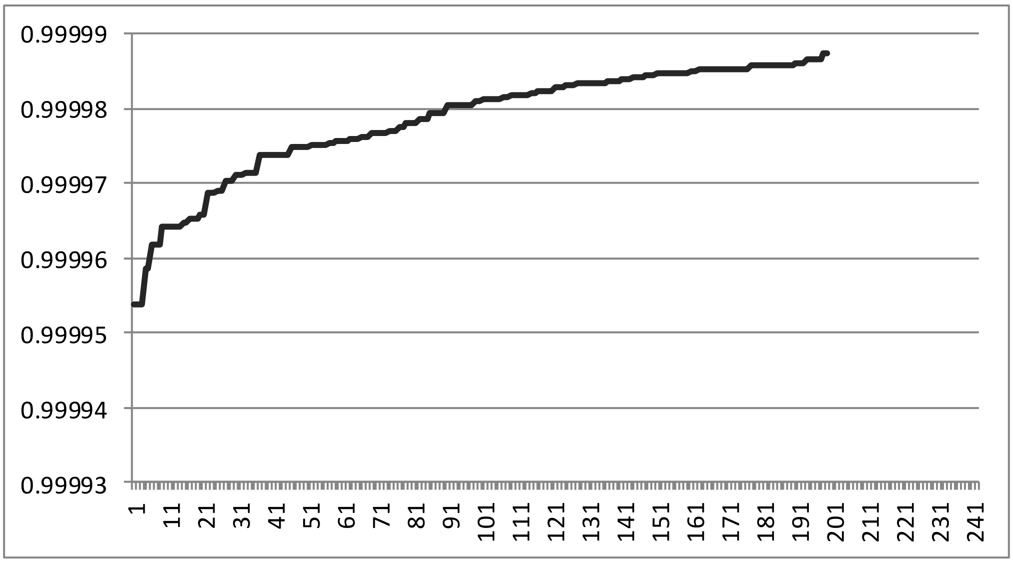


Figure 4.4: Convergence characteristics of single objective – reliability optimization in series system using proposed GA-PSO algorithm for 200 generations of a population (reliability VS Number of generation)

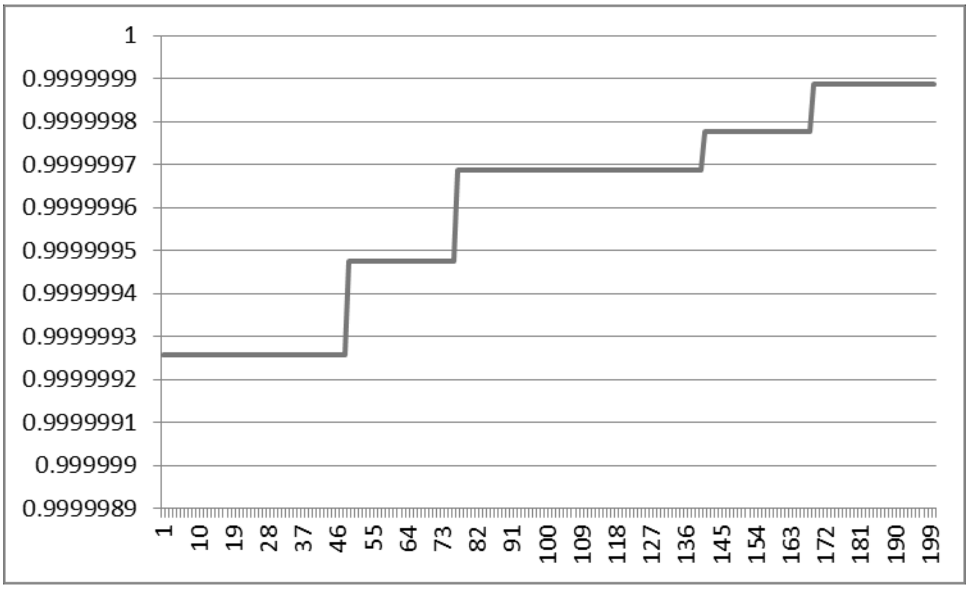


Figure 4.5: Convergence characteristics of single objective – reliability optimization in series-parallel system using proposed GA-PSO algorithm for 200 generations of a population (reliability VS Number of generation)

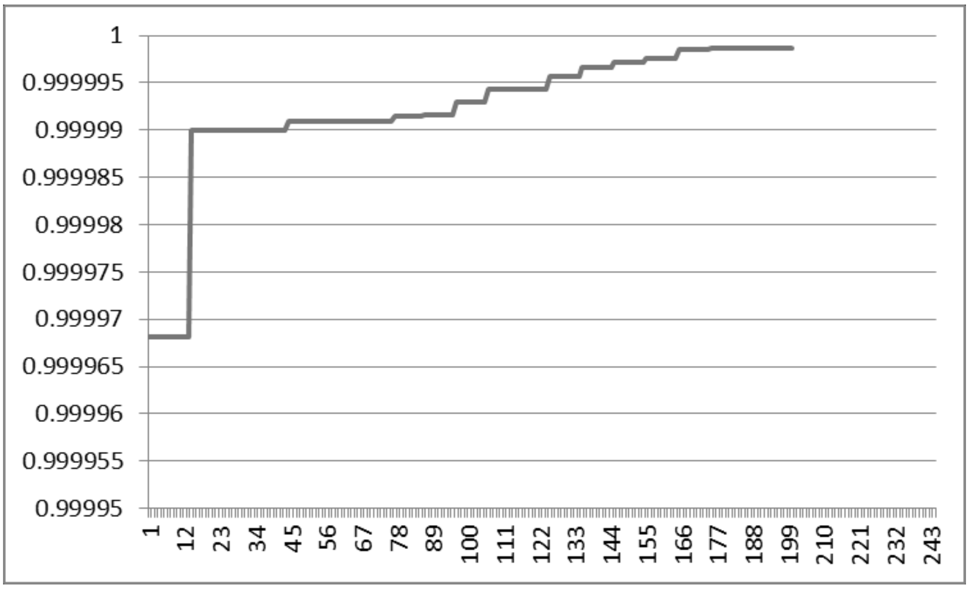


Figure 4.6: Convergence characteristics of single objective – reliability optimization in complex system using proposed GA-PSO algorithm for 200 generations of a population (reliability VS Number of generation)

4.5 Concluding Remarks

In this chapter, we have proposed an alternative hybrid approaches i.e., GA-PSO. In the case of GA-PSO, 50% of the chromosomes have been hybridized and 50% were used as particles in PSO-Co. In each iteration / generation of PSO, the particle best position is considered by comparing the GA population. On the other hand, the global best particle of PSO-Co is obtained by comparing both populations. Also, we have proposed a new crossover scheme i.e. power crossover. From the computational results, it is seen that the solutions obtained by the proposed approach are better than the same found by other heuristic and meta-heuristic algorithms reported in the literature for series, series-parallel and complex (bridge) systems. For further research, one can improve the proposed hybrid approach using advanced crossover, mutation operators for GA and velocity operator for PSO. Again, the proposed approach can be applied to solve other mixed-integer nonlinear programming problems.

CHAPTER 5

Multi-objective reliability optimization problem via Hybrid GA-PSO Algorithm

5.1 Introduction

The real-life design or decision making problems related to reliability optimization involve the simultaneous optimization of several objective functions. In the past as well as very recently, most of the researchers have been formulated by single objective optimization problems to solve the reliability optimization problems. For simultaneous maximization of system reliability and minimization of system cost of the reliability allocation problem, a multi-objective problem using a surrogate worth trade-off method [313] has been proposed by Sakawa. At the same time, Inagaki, Inoue, and Akashi [314] solved different problems by maximizing the system reliability and minimizing the system cost and weight by introducing an interactive optimization method. To focus further research in this area, one may recall the works of Park [315], Dhingra [316], Rao and Dhingra [317], Srinivas and Deb [318], Huang, Tian, and Zuo [319], Coit and Konak [320] and others. Taboada and Coit [321] proposed a new method which is based on the sequential combination of multi-objective evolutionary algorithms and data clustering on the prospective solutions. In the next year, Taboada, Espiritu, and Coit [322] developed an extension and applied a previously developed multi-objective evolutionary algorithm for solving the design allocation problems of multi-state series–parallel system for power system. Taboada, Espiritu, and Coit [323] solved multiple objective multi-state reliability optimization design problems by maximizing system reliability and minimizing both the system cost and weight. In the year 2009, Li, Liao, and Coit

[324] proposed a two-stage approach for multi-objective decision making with applications to system reliability optimization. Ramirez-Marquez and Rocco [325] developed a new evolutionary optimization technique for multi-state two-terminal reliability allocation in multi-objective problems. With the view to identify the combination of component failures that provide maximum reduction of network performance, Rocco, Ramirez-Marquez, Salazar, and Hernandez [326] studied the vulnerability analysis of a complex network. Dewri et. al [327] focused on optimal security hardening using multi-objective optimization on attack tree models of networks. Several researchers have solved reliability optimization problems considering single objective optimization problems described by Prasad & Kuo [328]; Levitin, Gregory, and Anatoly [329]; Kuo, Prasad, Tillman, & Hwang [330]).in their articles. It has been noticed that, the objective function(s) as well as constraints involving in reliability optimization problems, are non-convex and non-smoothness in nature.

In this chapter we have solved multi-objective optimization problem and for this purpose we have formulated multi-objective optimization problem considering objective functions as minimize the system cost, minimize the system volume and minimize the system weight respectively along with the restriction on targeted system reliability which is the only constraint of the problem. Here we have solved the problem using hybrid GA-PSO algorithm [204]. Finally, to find out the optimum result and to test the effectiveness of the GA-PSO algorithm, numerical example has been solved and computed results have been presented.

5.2 Formulation of Reliability-Redundancy Optimization Problems

The objective of the Reliability Redundancy Allocation problem is to enhance the reliability of the system under certain resource constraints based on the volume, cost and weight of the system. This objective can be fulfilled by adding an appropriate numbers of reliable redundant components. Mathematically reliability redundancy allocation problem has been formulated either single objective optimization problem or multi-objective optimization problem. In case of single objective optimization problem, the mathematical formulation is as follows:

$\text{Maximize}\, R_s = f(x_1, x_2, \ldots, x_n; r_1, r_2, \ldots, r_n)$

subject to $g_i(x_1, x_2, \ldots, x_n; r_1, r_2, \ldots, r_n) \le b_i$

where $x_i \in Z^+$, $0 \le r_i \le 1$ and $i = 1, 2, \ldots, m$

where R_s is the system reliability and $g_i(r, x)$ is the *i*-th constraint function, which are associated with system weight, volume and cost.

In case of multi-objective optimization problem the mathematical formulation is generally represented as follows:

$\text{Maximize}\, R_S(x, r)$

$\text{Minimize}\, \mathrm{C}_S(x, r)$

subject to $g_i(x_1, x_2, \ldots, x_n; r_1, r_2, \ldots, r_n) \le b_i$

where $x_i \in Z^+$, $0 \le r_i \le 1$ and $i = 1, 2, \ldots, m$

In this chapter we have formulated reliability redundancy allocation problem in terms of multi-objective optimization problem. The corresponding multi-objective optimization problem is as follows:

$\text{Minimize}\, \{\mathrm{V}_S(x), C_S(x, r), W_S(x)\}$

subject to $\mathrm{R}(x, r) \ge 0.999$

In our problem, we have considered five link complex (bridge) systems. The expression of $R(x, r)$ is

$$R_S(x,r) = R_1R_2 + R_3R_4 + R_1R_4R_5 - R_1R_2R_3R_4 - R_1R_2R_3R_5 - R_1R_3R_4R_5 - R_1R_2R_4R_5 + 2R_1R_2R_3R_4R_5$$

where $R_i(r_i, x_i) = 1 - (1 - r_i)^{x_i}$.

5.3 Multi-Objective Optimization

A multi-objective optimization problem can be written as

$\text{Minimize}\{f_1(x), f_2(x), \ldots, f_k(x)\}$

subject to, $x \in S$

where $f_i(x), i = 1, 2, \ldots, k$ is the i-th objective function and $S = \{x : g_j(x) \le 0, j = 1, 2, \ldots, m\}$ be the feasible region.

Before going to discuss about the solution procedure of multi-objective optimization problem, we shall define some useful definitions as follows:

Definition 1: A decision vector $x^* \in S$ is Pareto optimal if there does not exist another decision vector $x^* \in S$ such that $f_i(x) \le f_i(x^*)$ for at least one index $i, i = 1, 2, \ldots, k$.

Definition 2: A decision vector $x^* \in S$ is weakly Pareto optimal if there does not exist another decision vector $x \in S$ such that $f_i(x) \le f_i(x^*)$ for all $i = 1, 2, \ldots, k$.

Definition 3: A decision vector $x^* \in S$ is locally Pareto optimal if there does exist $\delta > 0$ such that x^* is Pareto optimal in $S \cap B(x^*, \delta)$ where $B(x^*, \delta)$ is an open ball with center at a point x^* and radius $\delta(> 0)$.

Definition 4: An objective vector minimizing each of the objective functions is called an ideal objective vector.

Definition 5: A utopian objective vector z^{**} is an infeasible objective vector whose components are formed by $z^{**} = z^* - \varepsilon_i$ for all $i = 1, 2, \ldots, k$ where z^* is the component of the ideal objective vector and $\varepsilon_i > 0$ is a relatively small but computationally significant scalar.

In the existing literature, several techniques/methods have been reported for solving the multi-objective optimization problems. In these techniques/methods, the multi-objective optimization problems have been formulated as single objective optimization problem. Some of these methods/techniques are as follows:

Problem 1: Global criteria method.

Problem 2: Weighted method.

Problem 3: ε-Constraint method.

Problem 4: Weighted sum method.

Problem 5: Tchebycheff method.

Problem 6: Weighted Tchebycheff method.

Problem 7: Lexicographic method.

Problem 8: Lexicographic Weighted Tchebycheff method.

All these methods are posteriori methods i.e., all these methods generated Pareto optimal solutions. For detail discussion about these problems one may refer to the book of K. Miettinen [331].

In this chapter, we have proposed an alternative method for solving multi-objective optimization problem. In this method, the transformed single objective optimization problem can be written as follows:

$$\text{Minimize} \sum_{i=1}^{k} w_i(\left| f_i(x) - f_i^* \right|) \tag{5.1}$$

subject to,

$$\sum_{i=1}^{k} w_i = 1$$

$$w_i \geq 0, i = 1, 2, \ldots, k$$

and $x \in S$

where f_i^* is the ideal objective value of the objective function $f_i(x)$.

This method looks like weighted approach introduced by Charnes and Cooper [332]. But slight difference is that we have taken deviation with respect to ideal objective vector instead of aspiration level from the goal.

5.4 Problem Formulation

Let us consider a complex bridge network system. In this system our objective is to minimize system volume, system cost and system weight subject to the targeted system reliability constraint. Therefore the problem can be written as

$$\text{Minimize}\{V_s, C_s, W_s\} \tag{5.2}$$

subject to $R_s \geq R^*$

where V_s, C_s, W_s are the system cost , system weight and system volume respectively and R^* is the targeted reliability which is fixed.

Hence the problem is to determine the number of redundant components together with reliability of each component by solving our proposed multi-objective optimization problem (5.2).

The above problem (5.2) can be written using our proposed problem as follows:

$$\text{Minimize}\{w_1(V_s - V_s^*) + w_2(C_s - C_s^*) + w_3(W_s - W_s^*)\} \tag{5.3}$$

subject to

$w_1 + w_2 + w_3 = 1$

$R_s \geq R^*$

$w_i \geq 0,\ i = 1,2,3$

where $V_s^*, C_s^* \text{ and } W_s^*$ be the components of ideal objective vector of $V_s^*, C_s^* \text{ and } W_s^*$. Clearly problem (5.3) is a single objective nonlinear constrained optimization problem. In this problem we have adjusted the weighted components by prior choices of objective functions.

According to different prior choices of weighted components, the following situations may arise.

Situation 1: Third constraint is more important that of second constraint and second constraint is more important that of first constraint i.e., $w_1 < w_2 < w_3$

Situation 2: Third constraint is more important that of first constraint and first constraint is more important that of second constraint i.e., $w_2 < w_1 < w_3$

Situation 3: Second constraint is more important that of third constraint and third constraint is more important that of first constraint i.e., $w_1 < w_3 < w_2$

Situation 4: Second constraint is more important that of first constraint and first constraint is more important that of third constraint i.e., $w_3 < w_1 < w_2$

Situation 5: first constraint is more important that of third constraint and third constraint is more important that of second constraint i.e., $w_2 < w_3 < w_1$

Situation 6: first constraint is more important that of second constraint and second constraint is more important that of third constraint i.e., $w_3 < w_2 < w_1$

For solving problem (5.3) we have used hybrid GA-PSO algorithm. Now we shall discuss the GA-PSO algorithm as follows:

5.5 Solution Procedures

To solve the problem mentioned in this chapter we have used hybrid GA-PSO algorithm. This hybrid algorithm is discussed in detail in the chapter 3.

5.6 Numerical Solutions, Results and Discussion

5.6.1 Numerical Examples

To illustrate the proposed approach for solving constrained multi-objective optimization problem by hybrid GA-PSO algorithm, the following numerical example has been considered.

$$\text{Minimize } V(x,r)=\sum_{j=1}^{5} v_j x_j^2$$

$$\text{Minimize } C(x,r)=\sum_{j=1}^{5} \alpha_j \left(\frac{-1000}{\ln r_j}\right)^{\beta_j} \left(x_j+\exp\left(\frac{x_j}{4}\right)\right)$$

$$\text{Minimize } W(x,r)=\sum_{j=1}^{5} wt_j x_j \exp(\frac{x_j}{4})$$

subject to $\mathrm{R}(x,r)\geq 0.999$

where

$$R_1(x,r)=R_1R_2+R_3R_4+R_1R_4R_5-R_1R_2R_3R_4-R_1R_2R_3R_5-R_1R_3R_4R_5$$

$$R_2(x,r)=2R_1R_2R_3R_4R_5-R_1R_2R_4R_5$$

$$R(x,r)=R_1(x,r)+R_2(x,r)$$

and $R_j(x_j,r_j)=1-(1-r_j)^{x_j}$

where $x_j \in Z^+$ and $0<r_j<1$. The parameters α_j and β_j are physical features of system components.

Now we have obtained ideal objectives v_s^*, c_s^*, w_s^* by solving three single objective optimization problems as follows:

For obtaining v_s^*, we have solved the following problem:

$$\text{Minimize } V(x,r)=\sum_{j=1}^{5} v_j x_j^2$$

subject to $\mathrm{R}(x,r)\geq 0.999$

For obtaining c_s^*, we have solved the following problem:

$$Minimize\ C(x,r)=\sum_{j=1}^{5}\alpha_j\left(\frac{-1000}{\ln(r_j)}\right)^{\beta_j}\left(x_j+\exp\left(\frac{x_j}{4}\right)\right)$$

subject to $\mathrm{R}(x,r)\geq 0.999$

For obtaining w_s^*, we have solved the following problem:

$$\text{Minimize } W(x,r)=\sum_{j=1}^{5}wt_jx_j\exp(\frac{x_j}{4})$$

subject to $\mathrm{R}(x,r)\geq 0.999$

Now we have solved the given multi-objective optimization problem by solving equivalent single objective optimization problem using our prosed technique/method.

Therefore the reduced single objective optimization problem is as follows:

$$Minimize\ w_1(\sum_{j=1}^{5}v_jx_j^2-10.0)+w_2(\sum_{j=1}^{5}\alpha_j\left(\frac{-1000}{\ln(r_j)}\right)^{\beta_j}\left(x_j+\exp\left(\frac{r_j}{4.0}\right)\right)-73.0)+$$

$$w_3(\sum_{j=1}^{5}\alpha_j\left(\frac{-1000}{\ln(r_j)}\right)^{\beta_j}\left(x_j+\exp\left(\frac{x_j}{4.0}\right)\right)-37.0)$$

where, $w_1+w_2+w_3=1$

and

$\mathrm{R}_s(x,r)\geq 0.999$

5.6.2 Experimental Results

For numerical experiments, we have considered reliability redundancy allocation problems of the complex (bridge) system. The values of all parameters related to the problem are given in Table 5.1. The proposed method has been coded in C programming language. The computational works have been performed on the PC having Intel core-2 duo processor with 3 GHz speed in Linux environment. In this computation, the values of parameters like size of population, maximum number of generation, crossover probability, and mutation probability have been taken as 200, 200, 0.90 and 0.10, respectively. In each case, 20 independent runs have been performed for each example to obtain the optimal/best found solution along with the corresponding value of the system reliability.

Table 5.1: Input parameters for complex system

Stage	$10^5\alpha$	β_i	v_i	wt_i	V	C	W
1	2.330	1.5	1	7	110	175	200
2	1.450	1.5	2	8			
3	0.541	1.5	3	8			
4	8.050	1.5	4	6			
5	1.950	1.5	2	9			

Table 5.2: Best found results for complex systems over 20 runs where = 0.5, = 0.3, = 0.2

category	x[1]	x[2]	x[3]	x[4]	x[5]	r[1]	r[2]	r[3]	r[4]	r[5]	R	Time
Min	4	1	2	1	1	0.885894	0.806363	0.753385	0.752838	0.756424	0.999899983	0.1874
Max	1	4	1	1	1	0.960728	0.835732	0.930935	0.83332 w_1	0.799392 w_2	0.99999994 w_3	0.2073
Mean											0.999933063	0.197996

Table 5.3: Best found results for complex systems over 20 runs where = 0.5, = 0.2, = 0.3

category	x[1]	x[2]	x[3]	x[4]	x[5]	r[1]	r[2]	r[3]	r[4]	r[5]	R	Time
Min	3	3	2	1	1	0.81665	0.820894	0.772895	0.888474	0.809925	0.999899983	0.2085
Max	2	1	1	3	1	0.753667	0.781214	0.775721	0.888115 w_1	0.776749 w_2	0.999998331 w_3	0.2038
Mean											0.999933468	0.195284

Table 5.4: Best found results for complex systems over 20 runs where = 0.3, = 0.5, = 0.2

category	x[1]	x[2]	x[3]	x[4]	x[5]	r[1]	r[2]	r[3]	r[4]	r[5]	R	Time
Min	2	3	1	1	1	0.927322	0.779439	0.809895	0.850323	0.77531	0.999899983	0.1854
Max	1	3	2	1	1	0.928402	0.788081	0.802895	0.955615 w_1	0.76506 w_2	0.99999994 w_3	0.1827
Mean											0.999939175	0.185876

Table 5.5: Best found results for complex systems over 20 runs where = 0.2, = 0.5, = 0.3

category	x[1]	x[2]	x[3]	x[4]	x[5]	r[1]	r[2]	r[3]	r[4] w_1	r[5] w_2	R w_3	Time
Min	3	1	1	1	2	0.772292	0.809056	0.89072	0.836044	0.785374	0.999899983	0.1805
Max	1	2	2	2	1	0.798974	0.790196	0.791197	0.772936	0.761983	0.999999523	0.1854
Mean											0.999941742	0.186518

Table 5.6: Best found results for complex systems over 20 runs where = 0.3, = 0.2, = 0.5

category	x[1]	x[2]	x[3]	x[4]	x[5]	r[1]	r[2]	r[3]	r[4] w_1	r[5] w_2	R w_3	Time
Min	2	1	2	1	1	0.994946	0.782743	0.820523	0.767712	0.809696	0.999899983	0.2119
Max	2	1	1	1	1	0.84643	0.940649	0.915134	0.831129	0.773882	0.999999523	0.1811
Mean											0.999926547	0.191848

Table 5.7: Best found results for complex systems over 20 runs where = 0.2, = 0.3, = 0.5

category	x[1]	x[2]	x[3]	x[4]	x[5]	r[1]	r[2]	r[3]	r[4] w_1	r[5] w_2	R w_3	Time
Min	4	1	1	2	1	0.965414	0.856815	0.837103	0.860352	0.87026	0.999899983	0.2043
Max	2	1	2	1	1	0.91368	0.784186	0.813336	0.886482	0.864216	0.999999464	0.2142
Mean											0.999944875	0.195212

Wilcoxon Rank-Sum Test

We have performed Wilcoxon Rank-Sum statistical test this chapter. It used to compare repeated measurements on a single sample to evaluate whether their populations mean ranks differ or not (paired difference test). We have used the paired difference test in this chapter, as because there is no other sample present to compare with the existing population.

Wilcoxon Rank-Sum Test for Complex System (Multi-objective reliability optimization)

Here we have referred population for population set generated by the algorithm using our proposed model when $w_1 = 0.5$, $w_2 = 0.3$, $w_3 = 0.2$.

Table 5.8: Ranks of objective-function value of generated population for complex system

Run	Objective Function value	Ranks of Objective Function value	Run	Objective Function value	Ranks of Objective Function value
1	0.999899983	3.5	11	0.99999994	17
2	0.999999940	17	12	0.99993306	9.5
3	0.999933063	9.5	13	0.99999994	17
4	0.99989998	3.5	14	0.99993306	9.5
5	0.99999994	17	15	0.99999994	17
6	0.99993306	9.5	16	0.99993306	9.5
7	0.99989998	3.5	17	0.999999523	13
8	0.99999994	17	18	0.99989998	3.5
9	0.99989998	3.5	19	0.99999994	17
10	0.99989998	3.5	20	0.99993306	9.5

Table 5.9: p-value calculation of the population for complex system

	Population 1
count	50
Rank-sum	210
α	0.05
W'	210
W-crit	337
median	0.999933063
Variance	$1.9285\text{X}10^{-09}$
Standard deviation	$4.39147\text{X}10^{-05}$

Conclusion: Since w'<w-crit therefore the population1 (sample) has significance which has been generated by the algorithm using our proposed model. We get value of w-crit using Wilcoxon Rank-Sum Table (http://www.real-statistics.com/statistics-tables/wilcoxon-rank-sum-table-independent-samples/).

Here we have referred population for population set generated by the algorithm using our proposed model when $w_1 = 0.5$, $w_2 = 0.2$, $w_3 = 0.3$.

Table 5.10: Ranks of objective-function value of generated population for complex system

Run	Objective Function value	Ranks of Objective Function value	Run	Objective Function value	Ranks of Objective Function value
1	0.99989998	4	11	0.99999833	17
2	0.99999833	17	12	0.99993347	10.5
3	0.99993347	10.5	13	0.99989998	4
4	0.99989998	4	14	0.99999833	17
5	0.99999833	17	15	0.99993347	10.5
6	0.99993347	10.5	16	0.99989998	4
7	0.99989998	4	17	0.99999833	17
8	0.99999833	17	18	0.99993347	10.5
9	0.99993347	10.5	19	0.99989998	4
10	0.99989998	4	20	0.99999833	17

Table 5.11: p-value calculation of the population for complex system

	Population 1
count	50
Rank-sum	210
α	0.05
W’	210
W-crit	337
median	0.999933468
Variance	$1.83616X10^{-09}$
Standard Deviation	$4.28504X10^{-05}$

Conclusion: Since w’<w-crit therefore the population1 (sample) has significance which has been generated by the algorithm using our proposed model. We get value of w-crit using Wilcoxon Rank-Sum Table (http://www.real-statistics.com/statistics-tables/wilcoxon-rank-sum-table-independent-samples/).

Here we have referred population for population set generated by the algorithm using our proposed model when $w_1 = 0.3$, $w_2 = 0.5$, $w_3 = 0.2$.

Table 5.13: Ranks of objective-function value of generated population for complex system

Run	Objective Function value	Ranks of Objective Function value	Run	Objective Function value	Ranks of Objective Function value
1	0.9991618	4	11	0.9998635	17
2	0.999004	19.5	12	0.99958515	12.5
3	0.9994278	9	13	0.99964714	4
4	0.9991417	4	14	0.99984741	17
5	0.9997616	14	15	0.99973774	12.5
6	0.99987993	10	16	0.99981403	4
7	0.9996662	4	17	0.99980927	17
8	0.99968162	19.5	18	0.99983788	8
9	0.9998691	11	19	0.99975392	4
10	0.9998094	4	20	0.99988964	15

Table 5.14: p-value calculation of the population for complex system

	Population 1
count	50
Rank-sum	210
α	0.05
W’	210
W-crit	337
median	0.999933468
Variance	$1.8745X10^{-09}$
Standard deviation	$4.32955X10^{-05}$

Conclusion: Since w’<w-crit therefore the population1 (sample) has significance which has been generated by the algorithm using our proposed model. We get value of w-crit using Wilcoxon Rank-Sum Table (http://www.real-statistics.com/statistics-tables/wilcoxon-rank-sum-table-independent-samples/).

Here we have referred population for population set generated by the algorithm using our proposed model when $w_1 = 0.2$, $w_2 = 0.5$, $w_3 = 0.3$.

Table 5.15: Ranks of objective-function value of generated population for complex system

Run	Objective Function value	Ranks of Objective Function value	Run	Objective Function value	Ranks of Objective Function value
1	0.99989998	4	11	0.99994174	10.5
2	0.99999952	17	12	0.99989998	4
3	0.99989998	4	13	0.99999952	17
4	0.99999952	17	14	0.99994174	10.5
5	0.99994174	10.5	15	0.99989998	4
6	0.99989998	4	16	0.99999952	17
7	0.99999952	17	17	0.99994174	10.5
8	0.99994174	10.5	18	0.99989998	4
9	0.99989998	4	19	0.99999952	17
10	0.99999952	17	20	0.99994174	10.5

Table 5.16: p-value calculation of the population for complex system

	Population 1
count	50
Rank-sum	210
α	0.05
W’	210
W-crit	337
median	0.999941742
Variance	$1.83938X10^{-09}$
Standard deviation	$4.2888X10^{-05}$

Conclusion: Since w’<w-crit therefore the population1 (sample) has significance which has been generated by the algorithm using our proposed model. We get value of w-crit using Wilcoxon Rank-Sum Table (http://www.real-statistics.com/statistics-tables/wilcoxon-rank-sum-table-independent-samples/).

Here we have referred population for population set generated by the algorithm using our proposed model when $w_1 = 0.3$, $w_2 = 0.2$, $w_3 = 0.5$.

Table 5.17: Ranks of objective-function value of generated population for complex system

Run	Objective Function value	Ranks of Objective Function value	Run	Objective Function value	Ranks of Objective Function value
1	0.99989998	3.5	11	0.99989998	3.5
2	0.99999952	17	12	0.99999952	17
3	0.99992655	9.5	13	0.99992655	9.5
4	0.99989998	3.5	14	0.99994174	13
5	0.99999952	17	15	0.99989998	3.5
6	0.99992655	9.5	16	0.99999952	17
7	0.99999952	17	17	0.99992655	9.5
8	0.99989998	3.5	18	0.99989998	3.5
9	0.99999952	17	19	0.99999952	17
10	0.99992655	9.5	20	0.99992655	9.5

Table 5.18: p-value calculation of the population for complex system

	Population 1
count	50
Rank-sum	210
α	0.05
W’	210
W-crit	337
median	0.999926547
Variance	$1.84326\text{X}10^{-09}$
Standard deviation	$4.29332\text{X}10^{-05}$

Conclusion: Since w’<w-crit therefore the population1 (sample) has significance which has been generated by the algorithm using our proposed model. We get value of w-crit using Wilcoxon Rank-Sum Table (http://www.real-statistics.com/statistics-tables/wilcoxon-rank-sum-table-independent-samples/).

Here we have referred population for population set generated by the algorithm using our proposed model when $w_1 = 0.3$, $w_2 = 0.2$, $w_3 = 0.5$.

Table 5.19: Ranks of objective-function value of generated population for complex system

Run	Objective Function value	Ranks of Objective Function value	Run	Objective Function value	Ranks of Objective Function value
1	0.99989998	4	11	0.99994488	10.5
2	0.99999946	17	12	0.99989998	4
3	0.99989998	4	13	0.99999946	17
4	0.99999946	17	14	0.99994488	10.5
5	0.99994488	10.5	15	0.99989998	4
6	0.99989998	4	16	0.99999946	17
7	0.99999946	17	17	0.99994488	10.5
8	0.99994488	10.5	18	0.99989998	4
9	0.99989998	4	19	0.99999946	17
10	0.99999946	17	20	0.99994488	10.5

Table 5.20: p-value calculation of the population for complex system

	Population 1
count	50
Rank-sum	210
α	0.05
W’	210
W-crit	337
median	0.999944875
Variance	$1.70082X10^{-09}$
Standard deviation	$4.1241X10^{-05}$

Conclusion: Since w’<w-crit therefore the population1 (sample) has significance which has been generated by the algorithm using our proposed model. We get value of w-crit using Wilcoxon Rank-Sum Table (http://www.real-statistics.com/statistics-tables/wilcoxon-rank-sum-table-independent-samples/).

5.7 Concluding Remarks

In this chapter, for the first time we have proposed a new method for solving constraint multi-objective optimization problem. There are several methods for solving constraint multi-objective optimization problem reported in the exiting literature. In the proposed method, the given multi-objective optimization problem has been transformed into single objective optimization problem. As all the objective functions and constraints are highly nonlinear so to solve this problem we have applied hybrid evolutionary algorithm. In case of reliability optimization, there are several ways to increase system reliability. These are i) to increase the reliability of each component ii) to use the parallel redundancy of the less reliable component iii) to use the stand by redundancy. As a result, consequently system cost, system volume and system weight also be increased. So to optimize the overall system considering system reliability, system cost, system volume and system weight the corresponding problem is formulated as multi-objective optimization problem. In this work we have considered minimization of system volume, minimization of system cost and minimization of system weight respectively along with the restriction on targeted system reliability which is the only one constraint of the problem. For solving the optimization problem we have used hybrid GA-PSO algorithm. In this algorithm we have applied GA for first 50% chromosomes and PSO for the rest. In each chromosome of GA and PSO, the first 50% genes are corresponding to integer variables and the remaining 50% genes corresponding to floating point variables. Finally, to illustrate the proposed method we have solved a numerical example and computational results have been presented. For future research one may apply the proposed method used in this chapter to solve multi-objective optimization problems in different areas of engineering, mathematics, economics and management science.

CHAPTER 6

Application of Reliability Redundancy Allocation Problem using hybrid GA-PSO in Wireless Sensor Networks (WSN)

6.1 Introduction

WSN consist of a sink node that works as server/gateway to the subnet of some inter-connected nodes, generally called motes [333] in a specified and similar environment to transmit data in interleaved fashion. The sink node supervises all dominant motes under the subnet. Each node consists of a tiny processor to process data, a storage system to store data, a power source to run the system uninterruptedly and a radio system to transmit data from one node to another. The various applications of WSN include different engineering environments, industrial process monitoring, battlefield surveillance, machine health monitoring, control systems etc. [334]. The development of efficient and reliable WSN faces different challenges like size, weight and cost (energy consumption) and others. There are some major constraints regarding resources such as energy consumption, memory size, computational speed and communication bandwidth. The basic goal of the reliability allocation problem is to measure the reliability levels of each component that either maximize the system reliability or minimize the system cost or both under several resources constraints. The Reliability Redundancy Allocation problem (RRAP) is basically a constrained, nonlinear mixed-integer programming problem. In this chapter, an example has been solved as nonlinear mixed-integer programming problem considering fuzzy parametric values. In real life situations, it has been observed that the reliability of an individual component may not be fixed. It may vary due to several reasons. There is no method/technology by which different components can be produced with exactly identical reliabilities. Also, the human factor, improper storage facilities and other environmental factors may affect the reliabilities of individual components. So, the reliability of each component is sensible and it may be treated as a positive imprecise number. To tackle the problems with such imprecise numbers, we have used fuzzy approach as it is more suitable and reliable to handle imprecise parameters.

6.2 The Reliability Redundancy Allocation Problem (RRAP) in Wireless Sensor Network (WSN) system

The Redundancy Allocation Problem is of interest when redundant components or parts are used to create a particular type of network. When similar components are used simultaneously one after another, i.e. serially, then the system is called series system, and when components are used in parallel then it is called a parallel system. The combination of series subsystems using parallel arrangement is called a series-parallel system. But the most important redundancy is seen in case of complex systems, systems with complex design, similar to WSN.

6.3 Solution Methodology

WSN can be best seen as complex bridge systems. In this chapter, the problem of Reliability Redundancy Allocation in WSN, in a maximization of Reliability in Redundancy Allocation formulation is approached using a new methodology. This chapter deals with the optimization of system reliability of redundancy allocation problem for complex (bridge) system with imprecise parameters. Different input variables (weight, volume and cost) are modelled as fuzzy numbers using triangular membership functions for three different linguistic categories likely Small (S), Medium (M) and High (H) respectively and inputs are fed in the hybrid evolutionary algorithm. Then this type of problems have been converted into non-linear fuzzy constraint optimization problem by using triangular fuzzy number and then the problem is converted into unconstrained optimization problems by Big-M penalty technique [298]. To solve the problems, we have proposed hybrid GAPSO algorithm for mixed-integer variables with tournament selection, intermediate crossover for integral values and power crossover for real values and one neighborhood mutation. The obtained fuzzy result set is linguistically classified into three categories, namely Satisfactory (S), Good (G) and Very Good (VG) respectively, for different combination of fuzzy input parameters. Here the decision making aspect is being imposed to the system and the system can deploy the decidability phenomena for different combinations of fuzzy input variables. Finally, the defuzzification is done to obtain precise values. Finally, to illustrate the theoretical development and also to test the performance of the proposed algorithm redundancy allocation problem for five link bridge network system has been solved for different representations of parameters and the simulation results have been compared.

6.4 Mathematical formulation of the Problem

The general form of the reliability optimization problem in crisp form is as follows:

Maximize $R_S(x,r)$ (6.1)

subject to

$g_i(x,r) \le b_i$, $i = 1, 2, \cdots, m$

where $(x,r) = (x_1, x_2, ..., x_n, r_1, r_{2,} ..., r_n)$,$1 \le l_j \le x_j \le u_j$, x_j isinteger, $j = 1, ..., n$,

$0 \le r_j \le 1, r_j$ isfloating point and b_i is the i-th available resource, $i = 1, 2, \cdots, m$.

If all the parameters are fuzzy valued, then the general form of the reliability optimization problem is

Maximize $\tilde{R}_S(x,\tilde{r})$ (6.2)

subject to

$g_i(x,\tilde{r}) \le \tilde{b}_i$, $i = 1, 2, \cdots, m$

where $x = (x_1, x_2, ..., x_n)$, $1 \le l_j \le x_j \le u_j$, x_j isinteger, j =1,..., n ,

$0 \le \tilde{r}_j \le 1, \tilde{r}_j$ isfloating point and $\tilde{b}_i$ is the i-th available resource which is imprecise , $i = 1, 2, \cdots, m$.

As the problem (6.2) is a constrained optimization problem, so we can solve the same by penalty function technique. In this technique, the constrained optimization problem is converted into unconstrained optimization problem.

Here we have used the Big-M penalty technique [298]. Hence, the unconstrained optimization problem corresponding to the problem (6.3) is as follows:

$$\text{Maximize } \tilde{R}_S(x,\tilde{r}) = \begin{cases} [\tilde{R}_S(x,\tilde{r})] & \text{when } x \in S \\ -\tilde{M} & \text{when } x \notin S \end{cases} \tag{6.3}$$

where $S = \{x : g_i(x,\tilde{r}) \le \tilde{b}_i,\ i = 1,2,\cdots,m \text{ and } 1 \le l_j \le x_j \le u_j,\ x_j \text{ is integer},\ j = 1,\ldots,n\}$.

6.5 Problem Description, Numerical Solutions and Results

Problem : Decision Making in Assessment of WSN RRAP using a Fuzzy-Hybrid Approach [224]

The Reliability Redundancy Allocation Problem (RRAP) in Wireless Sensor Network (WSN) systems is obviously an important problem. The basic function of a WSN system is to provide surveillance data transmission over a specified area maintaining minimum power consumption (minimum cost), occupying minimum volume and weight and having components with a reasonable level of reliability. In this chapter, a decision making assessment of Reliability Redundancy Allocation Problem (RRAP) is proposed using a fuzzy approach. The fuzzy approach uses the advantages of considering uncertainty in order to make the approach more practical. Triangular Fuzzy membership functions are used to encode fuzzy number sets as input variables (cost, weight and volume) to a hybrid optimization algorithm. A hybrid algorithm aiming for RRAP optimization of WSN system components is discussed. This algorithm is based on a new hybrid algorithm using a Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). The fuzzy results obtained are used to exhibit the decision making matrix, to enhance the decidability property of WSN. Finally after defuzzification, crisp data are obtained and compared with other approaches from literature and found satisfactory. In this problem, our main goal is to maximize the system reliability using a Hybrid algorithm to solve the Redundancy Allocation Problem with respect to different constraints.

Assumptions for the proposed system

(i) Reliability of each component is imprecise (fuzzy).

(ii) Failures of components are statistically independent.

(iii) The system will not be damaged or failed due to failed components.

(iv) All redundancies are active and there is no provision for repair.

(v) The components as well as the system have two different states, viz. operating state and failure state.

WSN is a combination of subsystems (nodes) and the arrangement of those subsystems can be considered as complex Systems. Even if each component/node is itself a set of other elements, reliability calculations will be treated as independent, characterized by specific reliability parameters [224]. The models of these subsystems/nodes ([1,2 ...X1], [1,2 ...X2] ...[1,2 ...X5]) are, in reliability terms, complicated systems with active reservation at element level, as depicted in Figure 6.1. Each component contains a number of serially connected elements. From the WSN reliability assessment viewpoint, there are two main approaches in the literature: a fuzzy approach, to make the problem more realistic, and optimization using a hybrid algorithm [204].

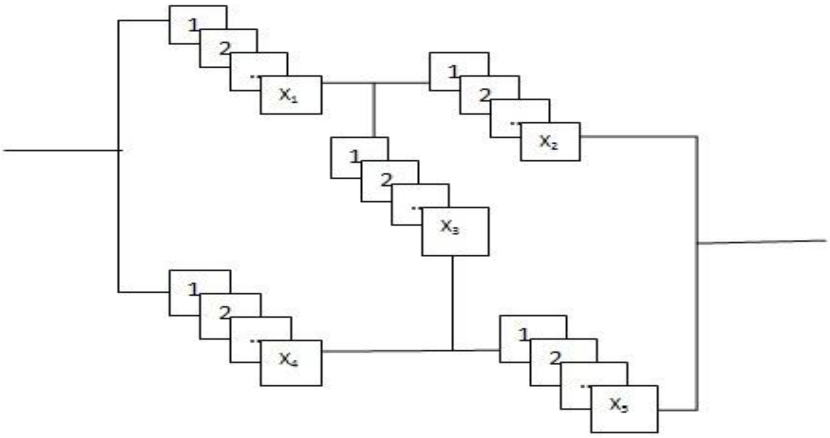


Figure 6.1: Structure of complex (bridge) WSN system

The corresponding optimization problem of complex systems is as follows:

$$\text{Maximize } f(r,x) = R_1R_2 + R_3R_4 + R_1R_4R_5 - R_1R_2R_3R_4 - R_1R_2R_3R_5 - R_1R_3R_4R_5 - R_1R_2R_4R_5 + 2R_1R_2R_3R_4R_5 \tag{6.4}$$

subject to:

$$g_1(x,r) = \sum_{j=1}^{5} v_j \cdot x_j^2 \le V \tag{6.5}$$

$$g_2(x,r) = \sum_{j=1}^{5} \alpha_j \cdot \left(\frac{-1000}{\ln r_j}\right)^{\beta_j} \left(x_j + \exp\left(\frac{x_j}{4}\right)\right) \le C \tag{6.6}$$

$$g_3(x,r) = \sum_{j=1}^{5} w_j \cdot x_j \cdot \exp\left(\frac{x_j}{4}\right) \le W \tag{6.7}$$

where $R_j(x_j, r_j) = 1-(1-r_j)^{x_j}$ and $x_j \in Z^+$ and $0 < r_j < 1$.

If all the parameters are fuzzy valued, then the general form of the reliability optimization problem is

$$\text{Maximize } f(r,x) = \tilde{R}_1 \times \tilde{R}_2 + \tilde{R}_3 \times \tilde{R}_4 + \tilde{R}_1 \times \tilde{R}_4 \times \tilde{R}_5 - \tilde{R}_1 \times \tilde{R}_2 \times \tilde{R}_3 \times \tilde{R}_4 - \tilde{R}_1 \times \tilde{R}_2 \times \tilde{R}_3 \times \tilde{R}_5 - \tilde{R}_1 \times \tilde{R}_3 \times \tilde{R}_4 \times \tilde{R}_5 \tag{6.8}$$

subject to:

$$g_1(x,\tilde{r}) = \sum_{j=1}^{5} \tilde{v}_j \cdot x_j^2 \le \tilde{V} \tag{6.9}$$

$$g_2(x,\tilde{r}) = \sum_{j=1}^{5} \alpha_j \cdot \left(\frac{-1000}{\ln \tilde{r}_j} \right)^{\beta_j} \left(x_j + \exp\left(\frac{x_j}{4} \right) \right) \le \tilde{C} \tag{6.10}$$

$$g_3(x,\tilde{r}) = \sum_{j=1}^{5} \tilde{w}_j \cdot x_j \cdot \exp\left(\frac{x_j}{4} \right) \le \tilde{W} \tag{6.11}$$

where $\tilde{R}_j(x_j,\tilde{r}_j) = (1-(1-\tilde{r}_j)^{x_j})$ and $x_j \in Z^+$ and $0 < \tilde{r}_j < 1$.

In this example, the complex system is considered with five sub-systems having system reliability *R₁,R₂,R₃,R₄* and *R₅* respectively. Following the process described in the previous section, the numerical solutions outcomes are described in different tables and result sets are constructed. Table 6.1 is constructed after applying fuzzification under three linguistic categories i.e. “Small”, “Medium” and “High”. Applying the meta-heuristic algorithms (i.e. GA, PSO and GA-PSO), fuzzy results is constructed in Table 6.2 over 50 runs. Also in Tables 6.3, 6.4 and 6.5, linguistic descriptions are given for complex systems over 50 runs using GA, PSO and GA-PSO algorithms respectively after linguistic classification. After defuzzification with the adaptive integration method, Table 6.6 presents the best results for the complex system over 50 runs, using the hybrid GA, PSO and GA-PSO algorithms. In Table 6.7, a comparison of the best GA, PSO and GA-PSO results with existing algorithms (for the complex system) is depicted. In Table 6.8, a statistical analysis for the complex system is presented, and, in Table 6.9, the average and standard deviation of CPU times (in seconds) over 50 runs are illustrated.

If for the cost (*C*), weight (W), and volume (V) the following linguistic categories: Small (S), Medium (M) and High (HL) are used, then the reliability of the systems can be characterized by a linguistic category of these input variables, as in Table 6.2 and Figures 6.2 up to 6.4. For reliability, other three linguistic categories can be defined: Satisfactory, Good, and Very Good. The choice of linguistic categories is based on the experience and data analysis.

The reliability will be determined for each linguistic category S, M, and H of the input variables (cost, weight and volume) using their fuzzy modeling. The calculations will be performed in each of the three breaking points of the triangular membership functions corresponding to the input variables (cost, weight and volume). Finally, the support for the assessment of reliability is based on Fuzzy Logic, which allows the establishment and using of fuzzy rules and fuzzification/defuzzification operations. Using a set of fuzzy rules such as "If ... Then", a decision table for the assessment of the reliability can be built.

Table 6.1: Input parameters for the complex system after fuzzification under three linguistic categories

Stage	$10^5\alpha$	β_i	v_i	w_i	V	C	W
1	2.330	1.5	1	7	[100-110 i.e. Small ,105-115 i.e. Medium,110-120 i.e. High]	[165-175 i.e. Small,170-180 i.e. Medium, 175-185 i.e. High]	[190-200 i.e. Small,195-205 i.e. Medium,200-210 i.e. High]
2	1.450	1.5	2	8			
3	0.541	1.5	3	8			
4	8.050	1.5	4	6			
5	1.950	1.5	2	9			

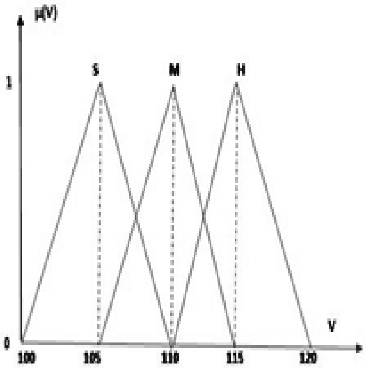


Figure 6.2: Triangular membership function (μ (V)) vs. Volume (V)

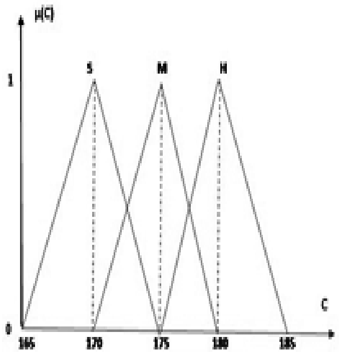


Figure 6.3: Triangular membership function (μ (C)) vs. Cost (C)

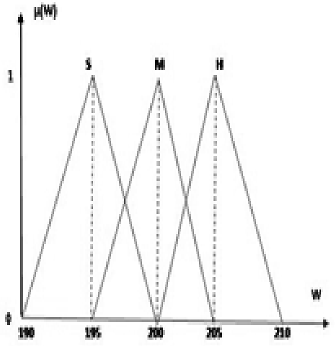


Figure 6.4: Triangular membership function (μ (W)) vs. Weight (W)

Table 6.2: Fuzzy Results for the complex system over 50 runs using the GA, PSO and GA-PSO algorithms before linguistic classification

Parameter	GA	PSO	GA-PSO
$f(r,x)$	[0.999954-0.999977 i.e. Satisfactory(S), 0.999966-0.999989 i.e. Good(G), 0.999977-0.999999 i.e. Very Good(VG)]	[0.9999961-0.9999981 i.e. Satisfactory(S), 0.9999971-0.9999991 i.e. Good(G), 0.9999981-0.999999 i.e. Very Good(VG)]	[0.99999951-0.99999971 i.e. Satisfactory(S), 0.99999961-0.99999981 i.e. Good(G), 0.99999971-1.00000000 i.e. Very Good(VG)]
x_1	4	4	4
x_2	3	3	3
x_3	3	3	3
x_4	3	3	3
x_5	1	1	1
r_1	[0.84909827-0.8511013, 0.8511014- 0.86513395, 0.86513396-0.8754369]	[0.84073487-0.84094575, 0.84094576-0.85068729, 0.85068730-0.85110130]	[0.8365171923-0.8371936873, 0.8371936874-0.8519672617, 0.8519672618-0.8525013013]
r_2	[0.85434493-0.85474791, 0.85474792- 0.86735129, 0.86735130,0.86765432]	[0.84568745-0.84436734, 0.84436735-0.85286560, 0.85286561-0.85474790]	[0.835183095-0.8357895612, 0.8357895613-0.8541784531, 0.8541784532-0.8547479179]
r_3	[0.91548745-0.91548745, 0.91548746-0.92168680, 0.92168681- 0.92568882]	[0.91434987-0.91588732, 0.91588733-0.92097698, 0.92097699-0.92168680]	[0.9051945876-0.9045762349, 0.9045762350-0.92098951627, 0.92098951628-0.9216868098]
r_4	[0.71262057-0.71463067, 0.71463068-0.72766142, 0.72766143-0.73786546]	[0.70564398-0.70716754, 0.70716755-0.71227634, 0.71227635-0.71262057]	[0.6955185123-0.6961839562, 0.6961839563-0.7119857657, 0.7119857658-0.7126205773]
r_5	[0.71519161-0.71559464, 0.71559465-0.71584979, 0.71584980-0.71860070]	[0.70765643-0.70876543, 0.70876544-0.71326765, 0.71326766-0.71559464]	[0.6916672354-0.6921904584, 0.6921904585-0.7149745328, 0.7149745329-0.7155946472]

Table 6.3: Linguistic Results for the complex system over 50 runs using the GA algorithm after linguistic classification

Reliability Analysis		Volume								
		Small			Medium			High		
	Weight	Small	Medium	High	Small	Medium	High	Small	Medium	High
Cost	Low	S	S	S	S	S	S	S	S	S
	Medium	S	S	S	S	S	G	S	G	G
	High	S	S	G	S	S	VG	S	G	VG

Table 6.4: Linguistic Results for the complex system over 50 runs using the PSO algorithm after linguistic classification

Reliability Analysis		Volume								
		Small			Medium			High		
	Weight	Small	Medium	High	Small	Medium	High	Small	Medium	High
Cost	Low	S	S	S	S	S	G	S	S	G
	Medium	S	S	G	S	G	G	S	G	G
	High	S	G	G	S	G	VG	G	G	VG

Table 6.5: Linguistic Results for the complex system over 50 runs using the GA-PSO algorithm after linguistic classification

Reliability Analysis		Volume								
		Small			Medium			High		
	Weight	Small	Medium	High	Small	Medium	High	Small	Medium	High
Cost	Low	S	S	S	S	S	G	S	G	G
	Medium	S	G	G	S	G	G	G	VG	VG
	High	S	G	VG	G	G	VG	G	VG	VG

Table 6.6: The best results for the complex system over 50 runs using the hybrid GA, PSO and GA-PSO algorithms after defuzzification

Parameters	The proposed GA algorithm	The proposed PSO algorithm	The proposed GA-PSO algorithm
$f(r,x)$	0.99999977	0.99999980	0.9999998085
x_1	4	4	4
x_2	3	3	3
x_3	3	3	3
x_4	3	3	3
x_5	1	1	1
r_1	0.86019260	0.84586730	0.8445448606
r_2	0.86102461	0.83795746	0.8449747568
r_3	0.91958763	0.91958763	0.9131117871
r_4	0.72319453	0.70942710	0.7040772028
r_5	0.71624539	0.71132103	0.7036067184

Table 6.7: Comparison of the best GA, PSO and GA-PSO results with existing algorithms (for complex system)

Parameters	Hybrid GA-PSO approach [306]	Efficient GA-PSO approach[204]	Proposed GA algorithm	Proposed PSO algorithm	Proposed GA-PSO algorithm
$f(r,x)$	0.99988964	0.99999952	0.99999977	0.99999980	0.9999998085
x_1	3	4	4	4	4
x_2	3	3	3	3	3
x_3	2	3	3	3	3
x_4	4	3	3	3	3
x_5	1	1	1	1	1
r_1	0.828134	0.858430	0.86019260	0.84586730	0.8445448606
r_2	0.857831	0.700000	0.86102461	0.83795746	0.8449747568
r_3	0.914192	0.922386	0.91958763	0.91958763	0.9131117871
r_4	0.648069	0.700000	0.72319453	0.70942710	0.7040772028
r_5	0.704476	0.700000	0.71624539	0.71132103	0.7036067184

Table 6.8: The statistical analysis for the complex system

	Maximum(Best)	Minimum(Worst)	Mean	Standard Deviation
Our proposed approach (GA)	0.999954	0.999853	0.999915	1.94656E-06
Our proposed approach (PSO)	0.9999970	0.99999096	0.999993505	1.86468E-06
Our proposed approach (GA-PSO)	0.9999998	0.999989	0.9999973	1.78658E-06
Sahoo et al.[204]	0.99999952	0.99998848	0.99999692	2.7907×10^{-6}
Sheikhalishahi et al. [279]	0.99988964	0.99988935	0.999889623	2.8226×10^{-11}

Table 6.9: Average and standard deviation of CPU times (in second) over 50 runs

	(Sheikhalishahi et al. [279])		(Sahoo et al. [204])		Our proposed approach (GA)		Our proposed approach (PSO)		Our proposed approach (GA-PSO)	
Problems	Average Time(s)	Standard Deviation (s)	Average Time(s)	Standard Deviation (s)	Average Time (s)	Standard Deviation (s)	Average Time(s)	Standard Deviation (s)	Average Time(s)	Standard Deviation (s)
Complexity	3.32	0.09	0.18	1.0×10^{-4}	0.15	1.0×10^{-4}	0.14	1.0×10^{-4}	0.12	1.0×10^{-4}

Wilcoxon Rank-Sum Test

This statistical test is used to compare two paired samples (populations) and to calculate the difference between each set of pairs and analyses these differences between matched samples. In this chapter we have performed the Wilcoxon Rank Sum statistical test to compare between two sample populations namely population1 and population2 and compared significance of one population over another one.

Wilcoxon Rank-Sum Test for Complex System (Reliability Redundancy Allocation Problem of WSN)

Here we have referred population1 for population set generated by algorithm using the proposed model of Sahoo et al.(2014) [204] and population2 for our proposed model.

Table 6.10: Ranks of objective-function value of different populations for complex system

Objective Function value			Ranks of Objective Function value		Objective Function value			Ranks of Objective Function value	
Run	population 1	population 2	Population1	Poplation2	Run	Population 1	Population2	Population 1	Population 2
1	0.9991618	0.999997616	5.5	90	26	0.99975681	0.999998635	23.5	95
2	0.999004	0.999989993	1	54.5	27	0.99986649	0.999995708	44.5	83
3	0.9994278	0.999996662	7.5	85.5	28	0.99982014	0.999990033	32.5	57
4	0.9991417	0.999968162	3.5	52	29	0.99975728	0.999990975	25	66.5
5	0.9997616	0.999998691	26.5	98	30	0.9991618	0.999990975	5.5	66.5
6	0.99987993	0.999990940	48.5	62	31	0.9998004	0.999998569	28	92
7	0.9996662	0.999998635	12.5	95	32	0.9994278	0.999992913	7.5	75
8	0.99968162	0.999999809	14.5	83	33	0.9991417	0.999994278	3.5	79
9	0.9998691	0.999990033	46.5	57	34	0.9997616	0.999997139	26.5	88
10	0.9998094	0.999990975	30	66.5	35	0.99987993	0.999991618	48.5	72.5
11	0.9998635	0.999999808	42.5	66.5	36	0.9996662	0.999990040	12.5	59.5
12	0.99958515	0.999998569	9	92	37	0.99968162	0.999994278	14.5	79
13	0.99964714	0.999992913	10	75	38	0.9998691	0.999991417	46.5	70.5
14	0.99984741	0.999994278	35	79	39	0.999094	0.999968162	2	52
15	0.99973774	0.999997139	16	88	40	0.9998635	0.999998691	42.5	98
16	0.99981403	0.999991618	31	72.5	41	0.99975392	0.999990940	21.5	62
17	0.99980927	0.999999809	29	59.5	42	0.99965668	0.999998635	11	95
18	0.99983788	0.999994278	34	79	43	0.99985695	0.999995708	37.5	83
19	0.99975392	0.999991417	21.5	70.5	44	0.99973791	0.999990033	17.5	57
20	0.99988964	0.999999520	50	100	45	0.99985759	0.999990975	40.5	66.5
21	0.99985695	0.999989993	37.5	54.5	46	0.99975204	0.999990975	19.5	66.5
22	0.99973791	0.999996662	17.5	85.5	47	0.99985695	0.999998569	37.5	92
23	0.99985759	0.999968162	40.5	52	48	0.99975681	0.999992913	23.5	75
24	0.99975204	0.999998691	19.5	98	49	0.99986649	0.999994278	44.5	79
25	0.99985695	0.999990940	37.5	62	50	0.99982014	0.999997139	32.5	88

Table 6.11: p-value calculation of different populations for complex system

	Population 1	Population 2
count	50	50
Rank-sum	3118	1932
α	0.05	0.05
W’	3118	NA
W"	NA	1932
Mean	0.999997527	0.999992978
Variance	$3.78789X10^{-12}$	$5.17448X10^{-11}$
Standard deviation	$1.97458X10^{-06}$	$7.19338X10^{-06}$
p-value	2	0

Conclusion: Since p-value of population2 (i.e., 0) is less than α value (i.e., 0.05) whereas p-value of population2 (i.e., 2) is greater than α value (i.e., 0.05), therefore the population 2 has significance over population1 in the generated sample (population).

The computational works have been performed on a PC with an Intel Core 2 Duo processor with 2.10 GHz speed and 1 GB RAM, in a Linux / Ubuntu platform. 50 independent runs have been made for the problem considering different sets of random fuzzy numbers using triangular fuzzy membership function. The proposed algorithm has been coded in the C++ / Matlab environment. In this simulation, a run is considered to be successful if the solution obtained is either the same or better than the known best found solution. From the overall analysis with respect to the best and worst found solutions, mean and standard deviation values of system reliabilities and also from the average and standard deviation of CPU times (in second), the proposed approach is the best compared to the existing approaches.

6.6 Conclusion

In this chapter, fuzzy input variables (cost, weight, and volume) have been introduced to the hybrid optimizer using triangular fuzzy membership function. After optimization, the result is classified by a linguistic classification method and ultimately the crisp value is obtained with the help of a defuzzification method, which is closer to the optimum value with respect to existing approaches. The proposed hybrid optimization technique can be improved by using advanced crossover, mutation, selection, and elitism operators for GA and position, velocity vector update operator for PSO. Probability of success, probability of failure and probability of repair are other considering factors, which creates an emerging scope to enhance the further development of this field.

CHAPTER 7

Reliability Optimization in Power Distribution Systems (PDS) using hybrid GA-ACO algorithm

7.1 Introduction

Electric Power Distribution Systems (PDS) are one of the most complex systems created by mankind. These systems include hundreds of thousands of components: transformers, distribution lines, generators, controls, protective equipment etc. In such a complex system, operation and planning conditions continuously vary. Constantly changing weather conditions (e.g., temperature wind speed), uncertainties and random factors make it extremely difficult to take correct decisions for smooth operation and planning of such systems. Wrong decisions cannot be corrected fast and they may result in substantial financial losses [335]. Many aspects of operation and planning of a PDS can be mathematically modelled as multi-objective optimization problems in which undesirable metrics such as costs, energy losses, errors etc. are to be minimized and desirable metrics such as profit, quality, reliability, energy efficiency etc. are to be maximized [336].

A PDS is a combination of different electrical devices, such as feeders, reclosers, switches, fuses etc. A feeder line is a route-peripheral in a power distribution network. It is used to connect load-points or consumers to the supply substations. Reclosers or Auto-reclosers, act as circuit breakers. When a fault occurs, a recloser causes the circuit to open automatically. Reclosers are used to detect and interrupt momentary faults in overhead power distribution systems. A switch is an electrical device to open or close connections in a PDS. A fuse acts as a low resistance device that causes opening of the circuit when there is some fault. Engineers need to choose the types, numbers and locations of these devices while designing a PDS. Among many ways to improve system performance, the ones based on the reliability indices are the most effective systems. System Average Interruption Frequency Index (SAIFI) and System Average Interruption Duration Index (SAIDI) are two such reliability indices which are directly connected to the performance [230] of a PDS. SAIFI is the "average number of interruptions that a customer would experience". SAIDI is the "average outage duration for each customer served" [227]. There are other reliability metrics too such as MAIFI ("average number of momentary interruptions that a customer would experience during a given period, typically a year") [228].

Reliability of a PDS may be improved by placing optimal number of switching and protective devices in a system. Computing optimal number of such devices and their positions with an objective of jointly minimizing the cost of the PDS and minimizing SAIFI and SAIDI reliability indices turns out to be a complex optimization problem. This chapter attempts to solve this problem using a hybridised evolutionary algorithm.

7.1.1 Objectives and contributions of this chapter

Most approaches in the literature are based on optimization technique, such as GA, ACO, PSO, DE, GSA etc. and for one single objective function. These methods are characterized by their own limitations; the possibility of mutual compensation of disadvantages when these methods are applied together remains unexplored. Although numerous papers have considered a wide range of heuristic techniques to solve different PDS optimization problems, to the authors' knowledge, none of them addresses the problem of optimal placement of switches and protective devices to improve reliability and minimize cost of a PDS using hybrid algorithms. The objective of this chapter is to examine the superiority of a hybrid algorithm to minimize different reliability indices (SAIFI and SAIDI) and the operational cost of a PDS in urban areas by choosing optimal number of protective devices (fuses, reclosers) and switches, and placing them optimally in a PDS. Here we have chosen an electric power distribution network of the Provincial Electricity Authority (PEA) of Thailand as a case study to demonstrate the performance of the proposed algorithm and compare the results with other existing solutions. Figure 7.1 describes the one-line diagram of the feeder. Length of each section, the number of customers and the average load of each section are presented in Table 7.3. Two search algorithms, namely GA and ACO, have been hybridized using new operators to solve single objective and multi-objective optimization problems in this context (see Table 7.4). Analyses of results have been shown in different tables (see Tables 7.5 – 7.15) and figures (Figure 7.3 – 7.7).

The rationale behind choosing GA and ACO as candidates for hybridization is as follows. GA is selected as a global search algorithm because of its robustness; GA is further improved upon by using a power-crossover operator [204] and a non-uniform mutation operator [337]. we are not using the elitism property of GA to induce greediness. This is because we are hybridizing GA with ACO, a known local search technique, which is also considered as "greedy". In ACO, ants probabilistically find optimal solutions by refining their trajectory in the local search space. As the refinement process has the property named "positive feedback for rapid detection of good solutions", ACO is fast and can adapt itself to changing situations [338]. So, we have chosen ACO as the local search technique.

We have also solved the same problem using GA and ACO individually. The results of GA, ACO and GA-ACO clearly indicate that the hybrid algorithm out performs the other two (See Table 7.14). We did not combine GA with Particle Swarm Optimization (PSO) in this chapter to solve the problem because both GA and PSO are global search-based techniques [339]. Although PSO can be adjusted to focus more on local search, such adjustments limit the speed or movement of particles [340]. Also, we have compared the results of the proposed algorithm with the hybrid GA-PSO [204].

In this chapter, we have demonstrated that the results obtained by the proposed GA-ACO algorithm are much better than the results reported in the literature for solving the same problem of finding optimum number of switches and their positions in a PDS. Thus, we have proposed an alternative design for more effective distribution network (see Table 7.4), which minimizes reliability indices and cost (Tables 7.5 – 7.13). Since lower values of SAIFI and SAIDI indicate less number of interruptions, the proposed network has reduced number of interruptions with respect to different protective and switching devices. Results also indicate that the performance of the proposed design is better than ones reported in the existing literature of Tippachon and Rerkpreedapong [240]. The major learning in this exercise is that hybrid algorithm combining GA and ACO is superior to individual GA and ACO for solving the problem at hand. It is also interesting to note that the operators used for the hybrid algorithm and the combination methodology has a positive impact on the performance of the hybrid algorithm. This is demonstrated by the fact that our combination performs better than the hybrid algorithm proposed in Lee et al. [334].

7.2 Problem formulation

7.2.1 Distribution Model

In this chapter, a simple but practical PDS [240] is considered as a case study. Any PDS can be described as a graph with devices as nodes and connections between devices as edges as shown in Figure 7.1. Considering the radial architecture of the network, any edge connecting two nodes n_i, n_j can be denoted using the number of the second node, n_j. To illustrate this, a few edges are numbered in Figure 7.1; (the remaining edges are not numbered to avoid confusion). Each edge in the graph of Figure 7.1 may be assumed to be a section and the label assigned to that edge is the index of that section. In Figure 7.1, there are 51 sections. Section-index represents a number that denotes the position of a section in a network. A “section-path” means a set of connecting section-indices starting from a section to the energy source. According to Levitin et al. [342], a section path for the Ith section can be represented as follows.

$$SEC_I = \{I, PRESEC\ (I), PRESEC\ (PRESEC\ (I)), PRESEC\ (\ PRESEC\ (PRESEC\ (I))\ \dots PRESEC^N(I)\} \quad (7.1)$$

PRESEC(I) denotes the predecessor function, which returns the predecessor section-index along the path to the energy source. In Figure 7.1, the section-path for the section-index 10 can be described as SEC_{10}= {1,5,7,8,10}.

A “load path” means a set of sections from a section containing a load point to the initial section. $LPATH_I$ is the load path containing all sections from the energy source to the load at the *I-th* section. Thus, a load-path can be represented as follows.

$$LPATH_I = \{I, PRESEC\ (I), PRESEC\ (PRESEC\ (I)), PRESEC\ (\ PRESEC\ (PRESEC\ (I))\ \ldots PRESEC^N (I)\} \quad (7.2)$$

In Figure 7.1, the load path $LPATH_{11}$ for the load point at Section 11 can be described as L_{11}= {1,5,7,8,9,11}.

In a PDS, three components, namely switches, fuses, and reclosers play greater role in enhancing reliability (Brown)[228]. That is why; these components have been chosen as major deciding factors to construct the proposed model in this chapter. Switches can isolate faulty parts of the distribution network and hence, they can help reduce customer interruption duration; however, they should not be considered as protective devices, like fuses or reclosers. Fuses protect the main feeder section (main section) separating the faulty lateral sections (shown by thin lines in Figure 7.1). Fuses are not allowed in the main section of the network because if a fault occurs, the main section will be disconnected from the network. A recloser is used as a protective, as well as a switching device. It handles permanent faults as well as temporary or sudden faults.

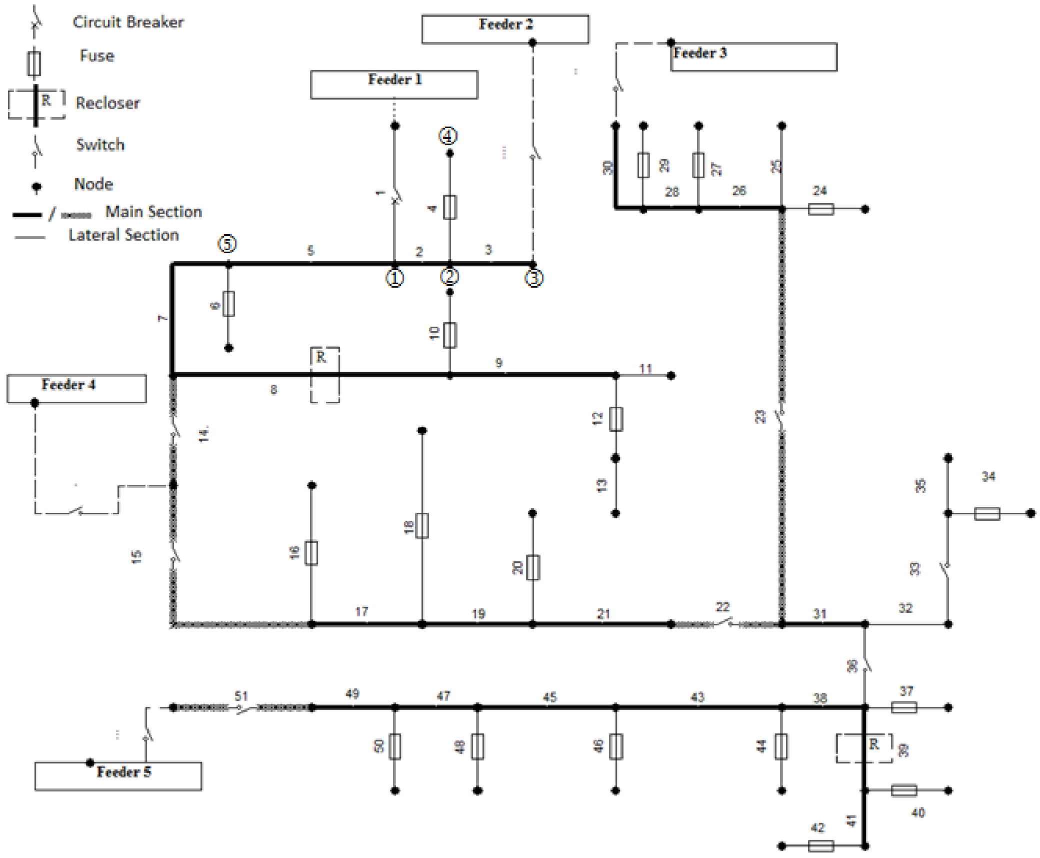


Figure 7.1: One-line diagram of a distribution feeder (22kV)

Whenever such a fault occurs, reclosers have unique ability to perform as an isolator to the downstream section so that the upstream section remains unaffected; this property is called trip/reclose function. This function protects the network against permanent faults.

7.2.2 Objective functions

This section presents single-objective as well as a multi-objective optimization problem for determining the optimal configuration of the devices in the PDS so that SAIFI, SAIDI and the system cost can be individually or jointly minimized. The total system cost consists of fixed cost and variable cost; fixed cost is incurred for installing the devices and variable cost is incurred during interruptions due to faults. Deviations of voltage and loads from normal permissible supply range are known as faults (Quiroga et al.) [343]. Faults can be temporary or permanent. A temporary fault occurs when protective systems allow the circuit to be reclosed after rectifying the fault in a short time period. Temporary faults include breakdown due to lightning, wind etc. Permanent faults means irreversible damage of components and it requires replacement of damaged components. Permanent faults include cable breakdown and damage of protective or switching devices. Electricity to customers in a PDS is supplied from substations. In urban areas, most electric substations supply electricity according to a consumption structure. The largest share goes to residential consumers (RES), followed by commercial consumers (COM). The remaining share of electricity goes to public (PUB) and the rest to few industrial consumers (IND). Universities (UNI), hospitals (HOS), public institutions (PINS), etc. are included in the public (PUB) category. In this chapter, five different customer types are considered. They are: residential – RES, University – UNI, Commercial – COM, Industrial – IND, and Supermarket – SMT. In the following equations, R, F and S represent the set of reclosers, fuses and switches along with their positions in the network; N and M represent total number of load-points and sections respectively. L_I is the load of the Ith load point and T represents the average duration of peak load, measured in hours per year. The value of T depends on the customer type. The values of T for five customer types described in this chapter are as follows: RES – 5840 Hours/Year, UNI –5660 Hours/Year, COM –4560 Hours/Year, IND –5110 Hours/Year, and SMT –7300 Hours/Year) [344].

λIJ is the permanent failure rate of the Ith load point due to the outage of the section J in the network. This failure rate is dependent on the topology of the circuit, arrangement of switches and protective devices (fuses and reclosers) and can be represented as follows.

$$\lambda_{IJ} = \begin{cases} \lambda_J & if\left(SEC_J \cap (R \cup F) - LPATH_I \cap (R \cup F)\right) = \emptyset \\ 0 & otherwise \end{cases} \tag{7.3}$$

λ_J denotes the permanent failure rate of the section *J*. In this paper, $\lambda_J$is assumed to be a random number in the permissible range. C_I is the number of customers at load-point *I*.

γ_{IJ} is the temporary failure rate of the I^{th} load-point due to an outage of the J^{th} section.

$$\gamma_{IJ} = \begin{cases} \gamma_J & if\left(SEC_J \cap R - LPATH_I \cap R\right) = \emptyset \\ 0 & otherwise \end{cases} \tag{7.4}$$

γ_J is the temporary failure rate of the J^{th} section. γ_J is assumed to be a random number in the permissible range. C_T is the interruption cost per kilowatt due to the temporary outage of load.

The Composite Customer Damage Function (CCDF) is the cumulative sum of the individual customer damage functions for different types of customers [240]. CCDF for different duration is presented in Table 7.2. In this chapter, C_T is set to CCDF depending on the duration of the fault.

r_{IJ} denotes the average interruption time of the load-point I due to a power outage of the section J . Power outage refers to power cut or power failure to the adjacent/ neighbouring sections. The neighbouring sections of section index 4 can be considered as section indices 2 and 3 in the network of Figure 7.1.

$$r_{IJ} = \begin{cases} r_{RPR} & if\left(SEC_J \cap S - LPATH_I \cap S\right) = \emptyset \\ & or\left(\overline{SEC_J} \cap S - \overline{LPATH}_I \cap S\right) = \emptyset \\ r_{SWT} & otherwise \end{cases}$$

(7.5)

$\overline{SEC_J}$ and $\overline{LPATH_I}$ are complement sets of SEC_J and $LPATH_I$ respectively. r_{RPR} and r_{SWT} represent repairing and switching time respectively whose values are properly set.

W_I is the annual energy consumption per customer, which can be represented as follows.

$$W_I = L_I T \tag{7.6}$$

7.2.2.1 Strategies for SAIFI

To represent this index, four strategies have been considered. These strategies are as follows.

The average frequency of permanent interruptions per customer$(i.e., SAIFI_1^C, f(R,F))$ [see Equation 7.7.1]

The average frequency of permanent interruptions per customer energy consumption $(i.e., SAIFI_2^W, f(R,F))$ [see Equation 7.7.2]

The arithmetic mean of i) and ii) $(i.e., SAIFI_3^A, f(R,F))$ [see Equation 7.7.3]

The geometric mean of i) and ii) $(i.e., SAIFI_3^G, f(R,F))$ [see Equation 7.7.4]

These indices can be represented as follows:[240].

$$SAIFI_1^C, f(R,F) = \frac{\sum_{I=1}^{N}(\sum_{J=1}^{M}\lambda_{IJ})C_I}{\sum_{I=1}^{N}C_I} \tag{7.7.1}$$

$$SAIFI_2^W, f(R,F) = \frac{\sum_{I=1}^{N}(\sum_{J=1}^{M}\lambda_{IJ})W_I}{\sum_{I=1}^{N}W_I} \tag{7.7.2}$$

$$SAIFI_3^A, f(R,F) = 0.5(\frac{\sum_{I=1}^{N}(\sum_{J=1}^{M}\lambda_{IJ})C_I}{\sum_{I=1}^{N}C_I} + \frac{\sum_{I=1}^{N}(\sum_{J=1}^{M}\lambda_{IJ})W_I}{\sum_{I=1}^{N}W_I}) \tag{7.7.3}$$

$$SAIFI_3^G, f(R,F) = \sqrt{\frac{\sum_{I=1}^{N}\left(\sum_{J=1}^{M}\lambda_{IJ}\right)C_I}{\sum_{I=1}^{N}C_I} * \frac{\sum_{I=1}^{N}\left(\sum_{J=1}^{M}\lambda_{IJ}\right)W_I}{\sum_{I=1}^{N}W_I}} \tag{7.7.4}$$

7.2.2.2 Strategies for SAIDI

This index is the average duration of interruption per customer and it is expressed in customer-minute unit. It represents the average time a customer is interrupted per year and it can be formulated as follows:[240].

To represent this index four strategies have been considered.

The average duration of interruption per customer $(i.e., SAIDI_1^C, f(R,F,S))$ [see Equation 7.8.1].

The average duration of interruption per customer energy consumption $(i.e., SAIDI_1^W, f(R,F,S))$ [see Equation 7.8.2]

The arithmetic mean of i) and ii) $(i.e., SAIDI_3^A, f(R,F,S))$ [see Equation 7.8.3]

The geometric mean of i) and ii) $(i.e., SAIDI_3^G, f(R,F,S))$ [see Equation 7.8.4]

$$SAIDI_1^C, f(R,F,S) = \frac{\sum_{I=1}^{N}(\sum_{J=1}^{M}\lambda_{IJ}r_{IJ})C_I}{\sum_{I=1}^{N}C_I} \tag{7.8.1}$$

$$SAIDI_1^W, f(R,F,S) = \frac{\sum_{I=1}^{N}(\sum_{J=1}^{M}\lambda_{IJ}r_{IJ})W_I}{\sum_{I=1}^{N}W_I} \tag{7.8.2}$$

$$SAIDI_3^A, f(R,F,S) = 0.5(\frac{\sum_{I=1}^{N}(\sum_{J=1}^{M}\lambda_{IJ}r_{IJ})C_I}{\sum_{I=1}^{N}C_I} + \frac{\sum_{I=1}^{N}(\sum_{J=1}^{M}\lambda_{IJ}r_{IJ})W_I}{\sum_{I=1}^{N}W_I}) \tag{7.8.3}$$

$$SAIDI_3^G, f(R,F,S) = \sqrt{\frac{\sum_{I=1}^{N}\left(\sum_{J=1}^{M}\lambda_{IJ}r_{IJ}\right)C_I}{\sum_{I=1}^{N}C_I} * \frac{\sum_{I=1}^{N}\left(\sum_{J=1}^{M}\lambda_{IJ}r_{IJ}\right)W_I}{\sum_{I=1}^{N}W_I}} \tag{7.8.4}$$

7.2.2.3 Strategies for cost function

This objective function is a combination of the fixed cost and the variable cost. The variable cost is dependent on two types of interruptions: interruption cost for permanent faults (PIC) and interruption cost for temporary faults (TIC) at different load points and sections of the network. In this chapter, switches and protective devices like reclosers and fuses are considered. *PIC$_{IJ}$ (TIC$_{IJ}$)* denotes *PIC (TIC)* of the Jth section with respect to Ith load point of the network.

The total cost is described as follows (Tippachon and Rerkpreedapong 2009).

$$COST_{TOTAL}, f(R,F,S) = COST_{FIXED} + \sum_{I=1}^{N}\sum_{J=1}^{M}(PIC_{IJ} + TIC_{IJ}) \tag{7.9}$$

$C_{IJ}(r_{IJ})$ is the interruption cost per kilowatt due to the permanent outage of the Ith load-point and the Jth section during r_{IJ}. $C_{IJ}(r_{IJ})$ can be represented as follows.

$$C_{IJ}(r_{IJ}) = (RES(\%) * f_R^{(r_{IJ})} +$$

$$UNI(\%) * f_U^{(r_{IJ})} + COM(\%) * f_I^{(r_{IJ})} + IND(\%) * f_S^{(r_{IJ})} + SMT(\%) * f_R^{(r_{IJ})}) \tag{7.10}$$

RES(%),UNI(%),COM(%), IND(%) and SMT(%) are percentages of different types of customers in the network and $f_R^{(r_{IJ})}$, $f_U^{(r_{IJ})}$,$f_I^{(r_{IJ})}$,$f_S^{(r_{IJ})}$are different cost functions for permanent interruptions. RES(%) is considered as 85%,UNI(%) as 0.2%, COM(%) as 13.5%, IND(%) as 1% and SMT(%) as 0.3% in an urban area (Neagu et al.) [344].

$\tilde{L}_I$ is the average load at the Ith load-point. *PIC$_{IJ}$* and *TIC$_{IJ}$* are given by the following equations.

$$PIC_{IJ} = C_{IJ}(r_{IJ})\tilde{L}_I\lambda_{IJ} \tag{7.11}$$

$$TIC_{IJ} = C_T\tilde{L}_I\gamma_{IJ} \tag{7.12}$$

7.3 Proposed Methodology

In this chapter, GA and ACO are hybridized after modifying some of their operators so that nonlinear problems can be solved. The modified GA and ACO algorithms are already described in chapter 3.

7.3.1 Chromosome Structure of the population

The chromosome structure (Figure 7.2) is considered to be a combination of the positions and numbers of Reclosers (R), Switches(S) and Fuses (F). The structure of a chromosome changes depending on the position and number of these devices. Initially, two reclosers at sections 8 and 39, five switches at sections 4,6,10,12,16, and six fuses at sections14,15,22,23,33,36 have been considered. The initial chromosome can be represented by {R(2)={8,39},S(5)={14,15,22,23,33},F(6)={4,6,10,12,16,18}}.

This representation has been chosen after observing the diagram of the existing system given in Figure 7.1. The upper limit of the number of devices has been fixed depending on the following conditions.

The maximum number of reclosers has been chosen as 4 as reclosers are very costly.

Fuses have been considered in each and every sub-section.

Switches have been considered at the beginning and at the end of every section points.The rest of the population are generated randomly with new random positions of reclosers, switches, and fuses.

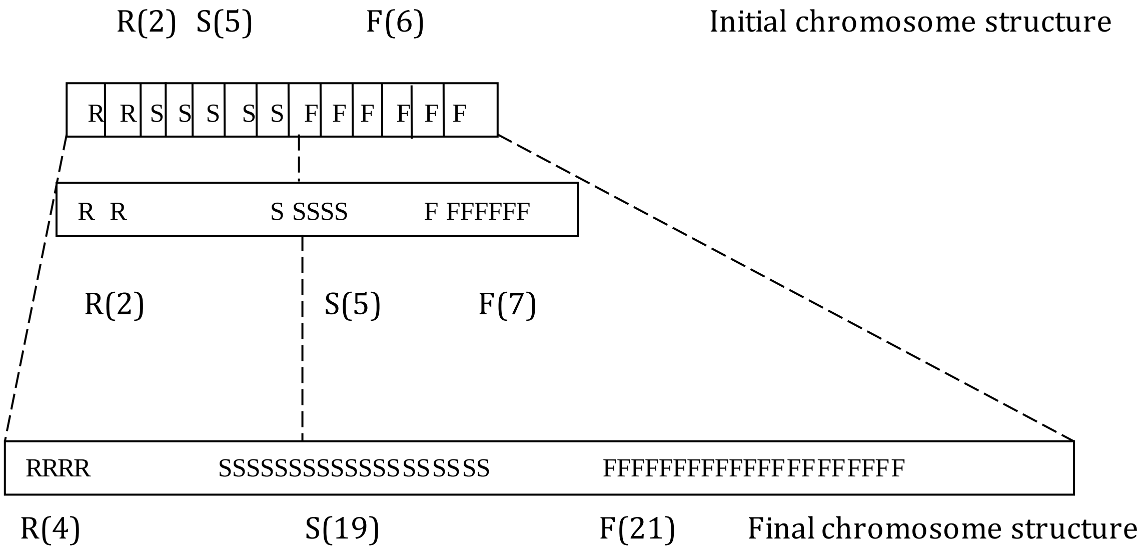

R(2) S(5) F(6)
Initial chromosome structure
R R S S S S S F F F F F F
R R S SSSS F FFFFFF
R(2)
S(5)
F(7)
RRRR SSSSSSSSSSSSSSSSSSS FFFFFFFFFFFFFFFFFFFFF
R(4)
S(19)
F(21)
Final chromosome structure

Figure 7.2: The chromosome structure of the population

For the multi-objective optimization problem solved in this chapter, the search space is three dimensional in nature. The first dimension (X-axis) represents the type of devices; the second dimension (Y-axis) represents the number of devices and the third dimension represents the location of different types of devices. For example A(R, 2, {8, 39}) represents a point that denotes 2 reclosers located at sections 8 and 39. A solution in this space is a triangle r that has three points whose X-coordinates are R, F and S. Let us call a point whose X-coordinate is R, F or S, an R-point, F-point or S-point respectively. The state transition rule is chosen to build a feasible solution (tour) to move from a triangle r (R1-point, F1-point, S1-point) to s(R2-point, F2-point, S2-point).

The hybrid GA-ACO Algorithm

To solve the problem mentioned in this chapter we have used hybrid GA-ACO algorithm. This hybrid algorithm has been discussed in detail in the chapter no 3.

7.4 Data for the GA-ACO Implementation for Optimal Placement of Protective and Switching Devices (OPPSD) in a PDS.

Table 7.1 presents the fixed cost required for installing protective devices in the network, and CCDF are provided in Table 7.2.Table 7.3 presents the Length (km), Average Load (kVA), Number of customers and Demand per customer in each section of the network.

Table 7.1: Fixed cost of protective equipment and switches [240]

Protective devices	Cost (USD)
Recloser	6000
Fuse	1500
Switch	2500

Table 7.2: Duration and Composite Customer Damage Function (CCDF) [240]

Duration (Interruption)	CCDF(USD/kW)
Temporary interruption	0.245
30 min	0.937
1.5hour	2.802

Table 7.3: Length (km), Average Load (kVA), Number of customers and Demand per customer in each section [240]

Section #	Length in Km	Load (Li) in kVA	Number of customers(Ni)	Demand per customer In kVA	Section #	Length in km	Load (Li) in kVA	Number of customers(Ni)	Demand per customer In kVA	Section #	Length in km	Load (Li) in kVA	Number of customers(Ni)	Demand per customer In kVA
1	3.4	0	0	NA	18	1.3	310	243	1.2757	35	0.5	100	30	3.3334
2	0.5	0	0	NA	19	2	340	147	2.3129	36	0.1	0	0	NA
3	0.1	0	0	NA	20	1.4	30	47	0.6383	37	2.5	100	65	1.5384
4	0.4	150	55	2.7272	21	1	2130	107	19.9065	38	3.2	480	50	9.6
5	0.5	0	0	NA	22	0.4	250	30	8.3333	39	0.8	500	1	500
6	1	220	89	2.4719	23	2.3	780	117	6.6667	40	3	450	220	2.0454
7	1	0	0	NA	24	4	610	135	4.5185	41	2.7	150	95	1.5789
8	3	1250	145	8.6207	25	0.7	80	95	0.8421	42	3	110	125	0.88
9	0.5	90	85	1.0588	26	1.5	60	50	1.2	43	9.3	60	167	0.3593
10	0.3	90	1	90	27	0.9	110	80	1.375	44	3.5	150	141	1.0638
11	1.2	445	200	2.225	28	4.2	590	120	4.9167	45	1.2	50	21	2.3809
12	1	720	2	360	29	0.7	90	93	0.9677	46	2	140	93	1.5053
13	1	30	55	0.5455	30	2.3	170	145	1.1724	47	0.9	36	106	0.3396
14	0.3	0	0	NA	31	2.8	480	65	7.3846	48	1.3	60	90	0.6667
15	2.9	150	55	2.7273	32	1.5	2400	80	30	49	1.3	30	17	1.7647
16	3	50	57	0.8772	33	1.3	210	102	2.0588	50	5	160	145	1.1034
17	1.7	60	105	0.5714	34	0.6	50	20	2.5	51	2	1350	67	20.1492

7.5 Simulation Results and Discussions

Simulation experiments have been carried out for computing three broad minimizations which are enumerated as follows.

I) Single objective – SAIFI minimization using the GA-ACO hybrid algorithm

A. $SAIFI_1^C$ minimization

B. $SAIFI_2^W$ minimization

C. $SAIFI_3^A$ minimization

D. $SAIFI_3^G$ minimization

II) Single objective – SAIDI minimization using the GA-ACO hybrid algorithm

A. $SAIDI_1^C$ minimization

B. $SAIDI_2^W$ minimization

C. $SAIDI_3^A$ minimization

D. $SAIDI_3^G$ minimization

III) Multi-objective – COST minimization using GA-ACO hybrid algorithm

Indices used in these three minimization categories have the following connotation. (C, 1) means that reliability indices are computed based on the number of customers; (W, 2) means that reliability indices are computed based on energy consumption; In minimizations where the subscript 3 is used, objective functions are combinations of two functions with subscript 1 and 2 and are computed as the arithmetic (A) or the geometric (G) mean.

SAIFI and SAIDI do not depend on each other; that is why, single objective minimization technique has been considered. The cost function is dependent on SAIFI and SAIDI; that is why, multi-objective minimization has been chosen for cost optimization.

Weighted sum multi-objective optimization technique has been used to obtain the multi-objective function. The multi-objective function is as follows.

$$f(Multi - Objective) =$$

$$SAIFI_1^C, f(R,F) + W_1 * SAIDI_1^C, f(R,F,S) + W_2 * COST_{TOTAL}, f(R,F,S) \qquad (7.13)$$

W_1, W_2 are weights assigned to $SAIDI_1^C$ and COST respectively. These coefficients represent the influence of the said parameters in the multi-objective function. As $SAIFI_1^C$ is measured as the number of interruptions and $SAIDI_1^C$ is measured as the duration of interruptions (in minute), weight of SAIDI_1^C is taken as 1.66X10^{-1}. W_2 is considered as 1.0X10^{-5} because the value of cost is of the order of 10^5. These weights transform the quantities in the range of small numbers.

The following assumptions are used to choose the positions and types of devices used in the distribution network.

- Each section contains only one device or none.
- An existing system containing 18 fuses, 2 reclosers, and 7 switches has been considered as a reference to obtain optimal values for the minimization problem.
- Only those simulation results have been considered, which produce better performance than the existing system.

7.5.1 Simulation Process

The proposed algorithm is implemented in C++ / Matlab and the experiment has been performed on Linux installed on a Pentium-dual-core PC having 2.10 GHz processor and 1 GB RAM. 50 independent runs have been carried out. In each run, the simulations are executed 50 times. To achieve a best possible solution for determining variables, different sets of random number have been considered for the problem. A run is considered successful if the result obtained is the same or better than the known "best-found" solution. In each run, either the solution converges or the number of iterations exceeds a given maximum number. Size of population on which GA or ACO takes place is set to 100. The maximum number of iterations for performing the evolution in a run is also set to 200. The probability of crossover between selected parents, pc is set to 0.9. The probability of mutation among selected offspring, p_m, is set to 0.1. The rate of evaporation for updating pheromone trail, ρ, is set to 0.1. The variable used to set condition for getting the position vector in Ant Colony System (ACS) is assigned a random number between 0 and 1. The average time to repair faults, r_{RPR} and average time to switch off the faulty section, r_{SWT}, are set to 2 hours and 0.5 hours respectively. β, weight of relative importance of heuristic information is set to 1.

After performing several experiments of the hybrid GA-ACO algorithm on nine minimization functions, the optimal number of devices and their positions in different sections of the network has been found out. These outcomes represent the optimal PDS which minimizes SAIDI or SAIFI or the COST. The systems so obtained after minimizing nine functions are presented in Table 7.4.

Table 7.4: Optimal PDSs obtained after minimizing nine objective functions

Case No.	System	Number of devices	Optimal position of devices in the section of the network

1	System proposed for best result in Single objective -$SAIFI_1^C$ minimization	{F17 , R4 , S7}	F17={4,6,12,16,18,20,24,25,27,29,37,40,42,44,46,48,50}
			R4 = {8,23,39,43}
			S7= {14,15,22,23,33,36,51}

2	System proposed for best result in Single objective -$SAIFI_2^W$ minimization	{F19 , R4 , S7}	F19 ={4,6,10,11,12,16,18,24,25,27,29,34,35,37,40,42,44,46,50}
			R4 = {8,23,39,43}
			S7= {14,15,22,23,33,36,51}

3	System proposed for best result in Single objective -$SAIFI_3^A$ minimization	{F18 , R4 , S7}	F18 ={4,6,11,13,16,18,20,24,25,27,29,37,40,42,44,46,48,50}
			R4 = {8,23,39,43}
			S7= {14,15,22,23,33,36,51}

4	System proposed for best result in Single objective -$SAIFI_3^G$ minimization	{F17 , R4 , S7}	F17={4,6,12,16,18,20,24,25,27,29,37,40,42,44,46,48,50}
			R4 = {8,23,39,43}
			S7= {14,15,22,23,32,36,51}

5	System proposed for best result in Single objective -$SAIDI_1^C$ minimization	{F18 , R4 , S16}	F18 ={4,6,10,13,16,18,20,24,25,27,29,37,40,42,44,46,48,50}
			R4 = {8,23,39,43}
			S16 = {9,15,17,19,21,22,26,28,30,31,32,33,38,41,47,51}

6	System proposed for best result in Single objective -$SAIDI_2^W$ minimization	{F19 , R4 , S16}	F19 ={4,6,10,11,12,16,18,24,25,27,29,34,35,37,40,42,44,46,50}
			R4 = {8,23,39,43}
			S16 = {9,15, 19,21,22,26,28,30,31,32,33,38,41,45, 49,51}

7	System proposed for best result in Single objective - $SAIDI_3^A$minimization	{F17 , R4 , S16}	F17={4,6,12,16,18,20,24,25,27,29,37,40,42,44,46,48,50}
			R4 = {8,23,39,43}
			S16 = {9,15,17,19,21,22,26,28,30,31,32,33,38,41,47,51}

8	System proposed for best result in Single objective -$SAIDI_3^G$ minimization	{F18 , R4 , S16}	F18 ={4,6,11,13,16,18,20,24,25,27,29,37,40,42,44,46,48,50}
			R4 = {8,23,39,43}
			S16 = {9,15,17,19,21,22,26,28,30,31,32,33,38,41,47,51}

9	System proposed for best result in Multi-objective –($SAIFI_1^C$,$SAIDI_1^C$ and $COST_{TOTAL}$) minimization	{F17 , R4 , S14}	F17={4,6,12,16,18,20,24,25,27,29,37,40,42,44,46,48,50}
			R4 = {8,23,39,43}
			S14 = {9,15,17,19,21, 26,28,30,31,32, 45,47,49,51}

7.5.2 Result Analysis

The results mentioned in Table 7.4 must be studied carefully. Recall that SAIFI, SAIDI and COST are dependent on λ_{IJ}, γ_{IJ}. $\lambda_{IJ}(\gamma_{IJ})$ is the permanent (temporary) failure rate of the Ith load point due to the outage of the section J. Interestingly, $\lambda_{IJ}(\gamma_{IJ})$ is dependent on $\lambda_J(\gamma_J)$, which is not a fixed value; rather, it depends on the duration of interruption (see Table 7.2). Thus, SAIFI, SAIDI and COST not only depend on the topology of the circuit (the number of reclosers, switches, fuses and their positions), but also on λ_J or γ_J, which is chosen as a random value in permissible range.

Thus, the optimum number and positions of fuses, switches and reclosers obtained after minimizing $SAIFI_1^C$as shown in the first row of Table 7.4 is just one outcome of some simulation runs and may change in another set of runs because the choice of λ_J and γ_J may be different in two separate simulation runs. The same is true for all rows in Table 7.4.

This motivates us to check whether the configuration of switches, fuses and reclosers mentioned as Case 1 in Table 7.4 truly minimizes $SAIFI_1^C$. So, we carried out another set of experiments in which the minimization of objective functions for $SAIFI_1^C$, $SAIDI_1^C$, and $(SAIFI_1^C, SAIDI_1^C, COST_{TOTAL})$ are executed again. During these runs, the value of $SAIFI_1^C$ is computed for PDS corresponding to cases 1 through case 9 as and when these configurations are encountered during the optimization process. If a configuration corresponding to a case is encountered a number of times, the minimum value of $SAIFI_1^C$ is preserved. These values are presented in Table 7.5. We note that the value of $SAIFI_1^C$ in the PDS corresponding to case 1 still remains the least as can be seen from Table 7.5.

Table 7.5: Result analysis of obtained values of $SAIFI_1^C$ from different case studies

Algorithms	Value of SAIFI$_1^C$								
Case No	01	02	03	04	05	06	07	08	09
single objective – $SAIFI_1^C$ minimization	3.8348	4.0091	3.9219	3.9209	4.5320	4.6482	4.5901	4.5897	4.7366
single objective – $SAIDI_1^C$ minimization	4.6615	5.3256	4.9935	4.9824	5.5483	5.7712	5.6597	5.6586	5.8876
Multi-objective – ($SAIFI_1^C$,$SAIDI_1^C$, $COST_{TOTAL}$ minimization	4.0754	4.4774	4.2764	4.2716	4.0658	4.8533	4.4595	4.4421	3.9202

In a similar manner, we check whether the configuration of switches, fuses and reclosers mentioned as Case 2 in Table 7.4 truly minimizes $SAIFI_2^W$. We carried out another set of experiments in which the minimization of objective functions for $SAIFI_2^W$ and $SAIDI_2^W$are executed again. During these runs, the value of $SAIFI_2^W$ is computed for PDS corresponding to cases 1 through case 9 as and when these configurations are encountered during the optimization process. If a configuration corresponding to a case is encountered a number of times, the minimum value of $SAIFI_2^W$ is remembered. These values are presented in Table 7.6. We note that the value of $SAIFI_2^W$ in the PDS corresponding to case 2 still remains the least as can be verified in Table 7.6.

Table 7.6: Result analysis of obtained values of $SAIFI_2^W$ from different case studies

Algorithms	Value of $SAIFI_2^W$								
Case No	01	02	03	04	05	06	07	08	09
single objective – $SAIFI_2^W$ minimization	3.4078	3.2597	3.3337	3.3329	3.8523	3.9511	3.9017	3.9013	4.0262
single objective – $SAIDI_2^W$ minimization	4.5269	3.9624	4.2446	4.2352	4.7162	4.9057	4.8109	4.8100	5.0046

Table 7.7 to Table 7.13 are prepared in a similar manner to reaffirm that the PDS corresponding to cases 3 to 9 actually minimizes the values of $SAIFI_3^A$, $SAIFI_3^G$, $SAIDI_1^C$, $SAIDI_2^W$, $SAIDI_3^A$, $SAIDI_3^G$, and $(SAIFI_1^C, SAIDI_1^C, COST_{TOTAL})$ respectively.

Table 7.7: Result analysis of obtained values of $SAIFI_3^A$ from different case studies

Algorithms	Value of $SAIFI_3^A$								
Case No	01	02	03	04	05	06	07	08	09
single objective – $SAIFI_3^A$ minimization	4.6497	4.7476	3.83481	4.4646	4.3810	4.5642	4.4726	4.4716	4.1986
single objective – $SAIDI_3^A$ minimization	5.4366	5.7712	4.66158	5.4273	5.3256	5.5482	5.4369	5.4357	5.1039

Table 7.8: Result analysis of obtained values of $SAIFI_3^G$ from different case studies

Algorithms	Value of $SAIFI_3^G$								
Case No	01	02	03	04	05	06	07	08	09
single objective – $SAIFI_3^G$ minimization	4.6556	4.7472	4.5640	3.6496	4.7476	4.1986	4.1625	4.6547	4.1625
single objective – $SAIDI_3^G$ minimization	5.6593	5.7707	5.5480	4.4364	5.7712	5.1038	5.0600	5.6582	5.0600

Table 7.9: Result analysis of obtained values of $SAIDI_1^C$ from different case studies

Algorithms	Value of $SAIDI_1^C$								
Case No	01	02	03	04	05	06	07	08	09
single objective – $SAIFI_1^C$ minimization	6.5083	6.4006	6.4544	6.4542	6.3790	6.2282	6.3036	6.3031	6.1851
single objective – $SAIDI_1^C$ minimization	4.0395	4.4204	4.2299	4.2256	3.7855	3.8442	3.8148	3.8147	3.9454
Multi-objective – ($SAIFI_1^C$,$SAIDI_1^C$, $COST_{TOTAL}$) minimization	4.6329	4.5562	4.5945	4.5943	4.5409	4.4335	4.4872	4.4868	4.4028

Table 7.10: Result analysis of obtained values of $SAIDI_2^W$ from different case studies

Algorithms	Value of $SAIDI_2^W$								
Case No	01	02	03	04	05	06	07	08	09
single objective – $SAIFI_2^W$ minimization	6.2328	6.1296	6.1812	6.4542	5.9645	6.1090	6.2093	6.2045	5.9233
single objective – $SAIDI_2^W$ minimization	3.8685	4.2333	4.0509	4.2256	3.6814	3.6252	3.9535	3.9441	3.7784

Table 7.11: Result analysis of obtained values of $SAIDI_3^A$ from different case studies

Algorithms	Value of $SAIDI_3^A$								
Case No	01	02	03	04	05	06	07	08	09
single objective – $SAIFI_3^A$ minimization	6.6582	6.7707	6.5480	6.4364	6.7712	6.1038	6.0600	6.7152	6.6593
single objective – $SAIDI_3^A$ minimization	4.7583	4.8529	4.6656	4.7308	4.3633	4.0471	4.0347	4.5613	4.7592

Table 7.12: Result analysis of obtained values of $SAIDI_3^G$from different case studies

Algorithms	Value of $SAIDI_3^G$								
Case No	01	02	03	04	05	06	07	08	09
single objective – $SAIFI_3^A$ minimization	6.6547	6.7472	6.5640	6.6496	6.7476	6.1986	6.1625	6.7016	6.6556
single objective – $SAIDI_3^A$minimization	4.6615	5.7712	5.7709	5.5482	5.3256	5.3254	5.3244	4.4366	5.3242

Table 7.13: Result analysis of obtained values of Cost from different case study

Algorithms	Value of Total Cost								
Case No	01	02	03	04	05	06	07	08	09
Multi-objective – ($SAIFI_1^C$,$SAIDI_1^C$,$COST_{TOTAL}$) minimization	147731	149335	149979	151590	153201	153846	155457	157068	146112

From these elaborate simulations as presented in Tables 7.5 to 7.8, we can conclude that the minimum values for SAIFI_1^C, SAIFI_2^W, SAIFI_3^A and SAIFI_3^G are obtained in case 1, case 2, case 3 and case 4 respectively. Similarly, from Tables 7.9 to 7.12, we can conclude that the minimum values of SAIDI_1^C, SAIDI_2^W, SAIDI_3^A and SAIDI_3^G are obtained in case 5, case 6, case 7 and case 8 respectively. Finally, from Table 7.13, we conclude that the minimum value for $\mathrm{COST}_{\mathrm{TOTAL}}$ is obtained in case 9.

7.5.3 Result comparison with other algorithms

Table 7.14 presents a comparative analysis of the values of SAIFI_1^C, SAIDI_1^C and COST for Single objective – SAIFI_1^C minimization, Single objective – SAIDI_1^C minimization and Multi-objective– COST minimization respectively by applying (i) ACS algorithm (Tippachon and Rerkpreedapong)[240], (ii) GA,(iii) ACO algorithm (iv) GA-PSO (Sahoo et al.)[204], (v) the proposed hybrid GA-ACO algorithm , and (vi) the GA –ACO algorithm due to Lee et al. [341]. From Table 7.14, it can be verified that the proposed hybrid GA-ACO algorithm minimizes reliability indices and cost better than ACS, GA, ACO, GA-PSO algorithms and the GA –ACO algorithm due to Lee et al. [334].

7.5.4 Statistical analysis

Table 7.15 presents a statistical analysis performed based on 50 runs for best results produced by the standard ACS algorithm (Tippachon and Rerkpreedapong)[240], GA algorithm, ACO algorithm, the GA-PSO algorithm, and the proposed GA-ACO algorithm and the GA –ACO algorithm due to Lee et al.

[341]. The maximum(worst), the minimum(best), mean values of all metrics, their standard deviations and elapsed time for the simulation runs are considered for different reliability indices(SAIFI, SAIDI) and cost minimization problems. These results are tabulated in Table 7.15. The best results are indicated using bold letters in the table.

7.5.5 Convergence Analysis

Figures 7.3 to 7.6 depict convergence characteristics of the proposed GA-ACO algorithm for single objective minimization problems (for reliability indices) and multi-objective minimization problem (for cost). We have considered 50 iterations. The diagrams show that the objective functions quickly converge towards global optima or become close to the minimum possible values starting from some initial value. For showing the convergence while minimizing the reliability indices, 2-Dimensional coordinate system has been used. Figures 7.7(A) and 7.7(B) depict the convergence characteristics of multi-objective ($\mathrm{SAIFI}_1^{\mathrm{C}}$, $\mathrm{SAIDI}_1^{\mathrm{C}}$, $\mathrm{COST}_{\mathrm{TOTAL}}$) minimization using the proposed GA-ACO algorithm. Figure 7.7(A) is shown in 3-Dimentional coordinate system; X-axis denotes $\mathrm{SAIFI}_1^{\mathrm{C}}$; Y-axis denotes the number of iterations and Z-axis denotes $\mathrm{COST}_{\mathrm{TOTAL}}$. In Figure 7.7(B) X-axis denotes $\mathrm{SAIDI}_1^{\mathrm{C}}$; Y-axis denotes the number of iterations and Z-axis denotes $\mathrm{COST}_{\mathrm{TOTAL}}$.

Figures 7.3 – 7.6 show convergence characteristics of single objective functions. Figures 7.7(A) and 7.7(B) show convergence characteristics of the multi-objective function. In case of single objective minimizations, the objective function is dependent only on the number of iteration; so, it is 2-Dimentional in nature. On the other hand, for multi-objective minimization, the objective function is dependent on three factors, $\mathrm{SAIFI}_1^{\mathrm{C}}$, $\mathrm{SAIDI}_1^{\mathrm{C}}$,$\mathrm{COST}_{\mathrm{TOTAL}}$ (see Equation 7.13). So, we used two 3-Dimensioanl graphs in this case.

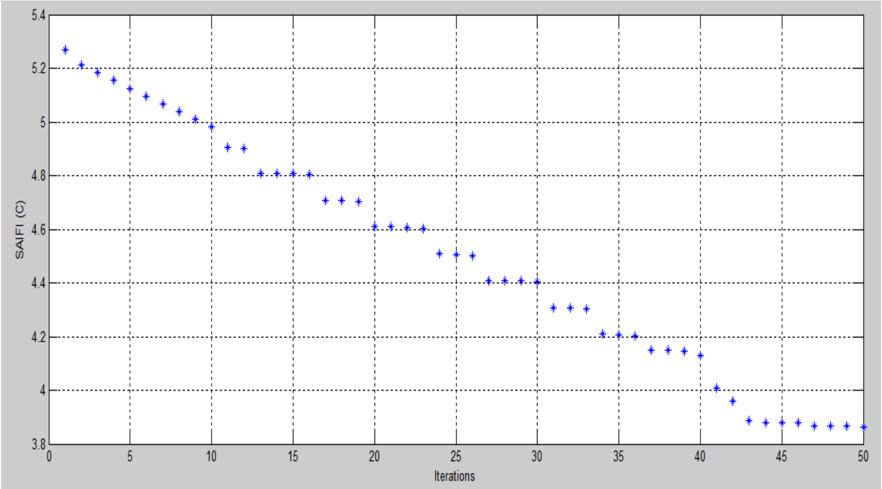


Figure 7.3: Convergence characteristics of single objective – SAIFI_1^C minimization using proposed GA-ACO algorithm for 50 iterations

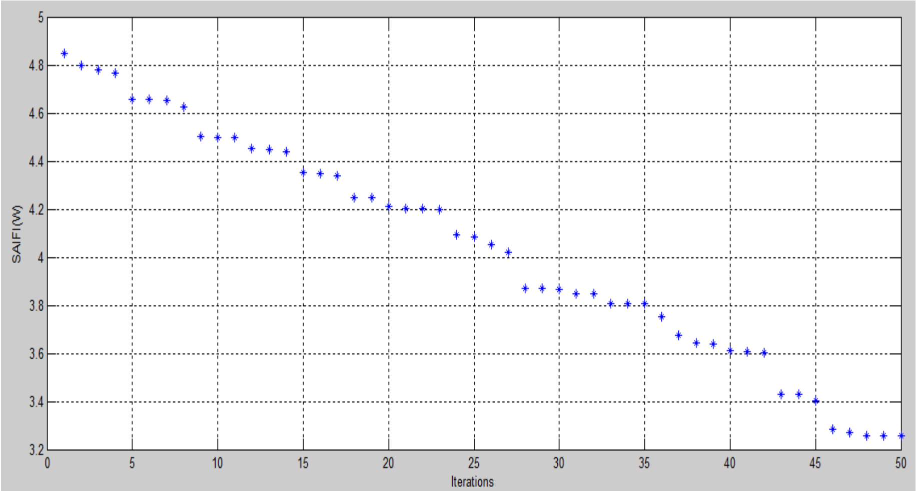


Figure 7.4: Convergence characteristics of single objective – SAIFI_2^W minimization using proposed GA-ACO algorithm 50 iterations

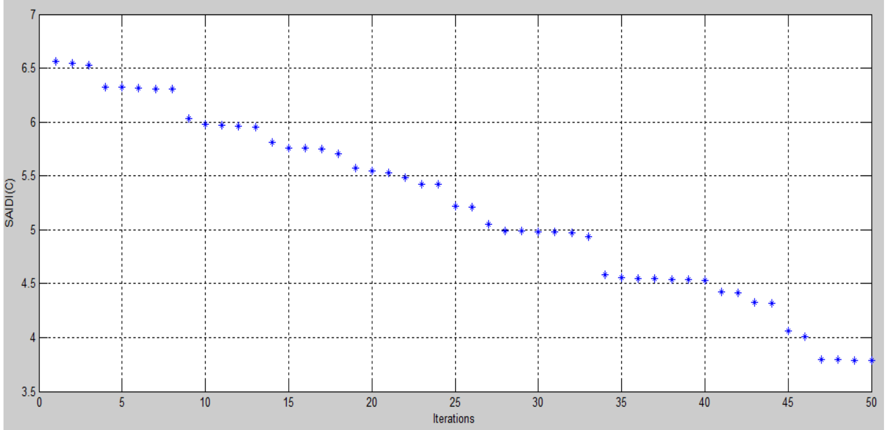


Figure 7.5: Convergence characteristics of single objective – SAIDI_1^C minimization using the proposed GA-ACO algorithm 50 iterations

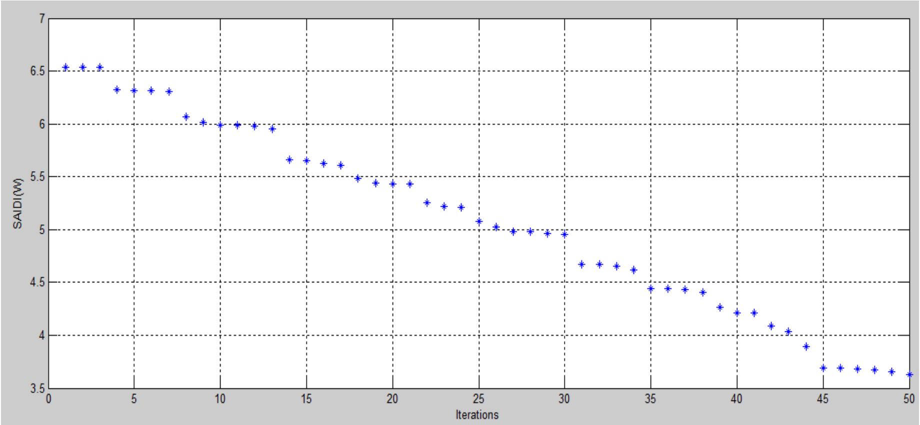


Figure 7.6: Convergence characteristics of single objective –SAIDI_2^W minimization using the proposed GA-ACO algorithm 50 iterations

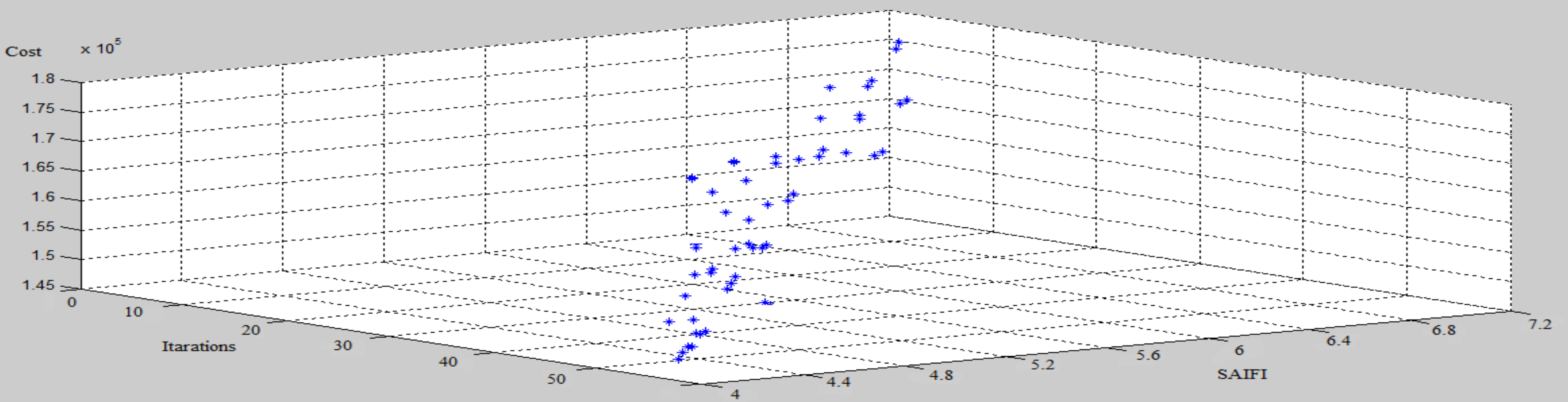


Figure 7.7A: Convergence characteristics (SAIFI vs Iterations vs Cost) of Multi-objective – ($SAIFI_1^C$,$SAIDI_1^C$, $COST_{TOTAL}$) minimization using the proposed GA-ACO algorithm in 50 runs

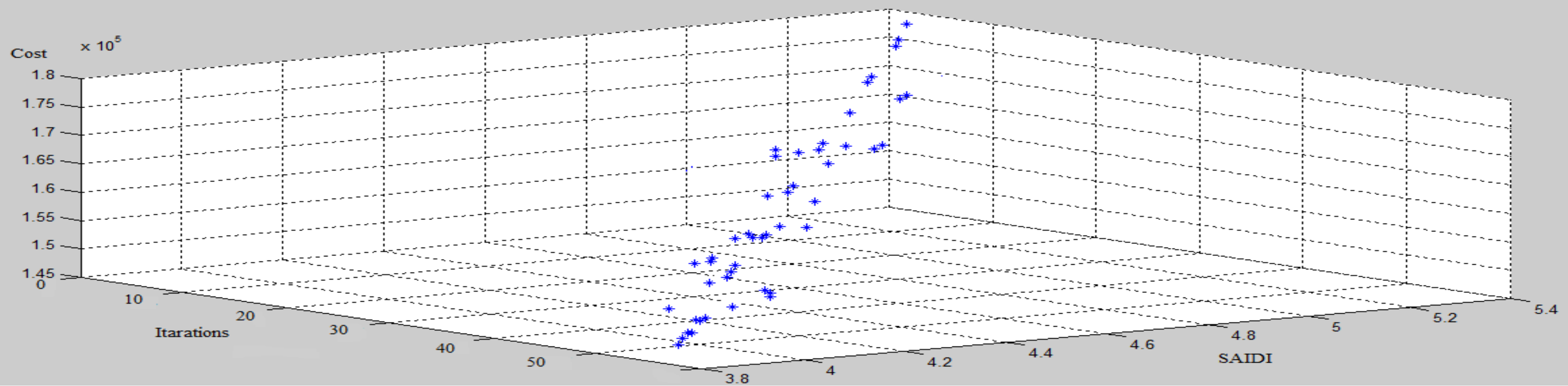


Figure 7.7B: Convergence characteristics (SAIDI vs Iterations vs Cost) of Multi-objective – ($SAIFI_1^C$,$SAIDI_1^C$, $COST_{TOTAL}$) minimization using the proposed GA-ACO algorithm in 50 runs

Table 7.14: Comparison of results obtained by the standard ACS algorithm (Tippachon and Rerkpreedapong)*, GA, ACO algorithm, GA-PSO algorithm (Sahoo et al.)[204]**, the proposed GA-ACO algorithm and GA-ACO algorithm (Lee et al.) [341]*** using our operators.

Systems	I. Result from existing literature* II. Result from proposed GA algorithms.III. Result from proposed ACO algorithm IV. Result from existing** GA-PSO V. Result from proposed GA-ACO algorithm. VI. Result from existing*** GA-ACO(Lee et al.)[335][341] using our operators	Value of $SAIFI_1^C$	Improvement in percentage compared to existing literature[240]	Value of $SAIDI_2^C$	Improvement in percentage compared to existing literature[240]	Value of Total Cost	Improvement in percentage compared to existing literature[240]
Existing System without any minimization	NA	8.6712	NA	10.254	NA	NA	NA
Single objective – $SAIFI_1^C$ minimization	I	4.0784	NA	6.7213	NA	NA	NA
	II	7.1402	NA	9.0763	NA	NA	NA
	III	5.6093	NA	6.5501	NA	NA	NA
	IV	6.3747	NA	7.8988	NA	NA	NA
	V	3.8343	5.98%	6.3790	5.09%	NA	NA
	VI	3.95635	2.992%	6.461575	3.86%	NA	NA
Single objective – $SAIDI_2^C$ minimization	I	4.8658	NA	4.0255	NA	NA	NA
	II	7.5389	NA	8.0978	NA	NA	NA
	III	5.0701	NA	4.2655	NA	NA	NA
	IV	6.2023	NA	5.9416	NA	NA	NA
	V	4.6615	4.2%	3.7855	5.96%	NA	NA
	VI	4.76365	2.099%	3.9055	2.98%	NA	NA
Multi-objective optimization	I	4.1593	NA	4.8454	NA	159,900	NA
	II	7.5198	NA	8.4511	NA	236,212	NA
	III	4.2788	NA	5.0667	NA	185,337	NA
	IV	6.0158	NA	6.6482	NA	198,055	NA
	V	3.9202	5.75%	4.4028	9.13%	146,112	8.62%
	VI	4.03975	2.5%	4.6241	4.567%	153,006	4.311%

Table 7.15: Statistical analysis for the best results (mean, standard deviation etc.) produced by I) the standard ACS algorithm (Tippachon and Rerkpreedapong)[240]*, II) GA, III) ACO algorithm, IV) GA-PSO algorithm (Sahoo et al.)[204]**, and V) the proposed GA-ACO algorithm.

Type	I. Result from existing literature* II. Result from proposed GA algorithms. III. Result from proposed ACO algorithm IV. Result from existing** GA-PSO V. Result from proposed GA-ACO algorithm	Parameters	Maximum (Worst)	Minimum (Best)	Mean	Standard deviation	Average Elapsed Time (in second)
Single objective SAIFI minimization	I	$SAIFI_1^C$	5.6039	4.0784	5.1064	0.953665	0.893858
	II	$SAIFI_1^C$	9.8109	7.1402	8.4221	0.787476	0.867856
		$SAIFI_2^W$	9.7769	6.5656	8.1712	0.954786	
	III	$SAIFI_1^C$	7.7074	5.6093	6.0642	0.657044	0.569863
		$SAIFI_2^W$	7.4972	5.0347	6.4969	0.731636	
	IV	$SAIFI_1^C$	8.7591	6.3747	7.5142	0.731568	0.564789
		$SAIFI_2^W$	8.6370	5.8001	7.2698	0.843255	
	V	$SAIFI_1^C$	5.2685	3.8343	4.76653	0.727777	0.413807
		$SAIFI_2^W$	4.8541	3.2597	3.73802	0.818183	
Single objective SAIDI minimization	I	$SAIDI_1^C$	10.0218	6.7213	8.3287	0.991928	0.548769
	II	$SAIDI_1^C$	10.2368	9.0763	9.5490	0.376595	0.764593
		$SAIDI_2^W$	10.1943	8.916	9.5040	0.35792	
	III	$SAIDI_1^C$	8.2940	6.5501	7.3871	0.518499	0.673807
		$SAIDI_2^W$	9.5276	6.3898	8.0214	0.932935	
	IV	$SAIDI_1^C$	8.6204	7.8988	7.8331	8.021456	0.876390
		$SAIDI_2^W$	9.9535	7.7385	8.7574	0.658567	

Type	I. Result from existing literature* II. Result from proposed GA algorithms. III. Result from proposed ACO algorithm IV. Result from existing** GA-PSO V. Result from proposed GA-ACO algorithm	Parameters	Maximum (Worst)	Minimum (Best)	Mean	Standard deviation	Average Elapsed Time (in second)
	V	$SAIDI_1^C$	6.5276	3.7855	5.15655	1.37105	0.430800
		$SAIDI_2^W$	6.5367	3.6252	5.08095	1.45575	
Multi-objective ($SAIFI_1^C$, $SAIDI_2^C$ and $COST$)minimization	I	$SAIFI_1^C$	5.6803	4.1593	4.8589	0.452226	0.875432
		$SAIDI_2^C$	7.5325	4.8454	6.0814	0.798932	
		COST	192252	159,900	174,782	9805.73	
	II	$SAIFI_1^C$	9.2698	7.5198	8.3248	0.520312	0.459864
		$SAIDI_2^C$	10.1379	8.4511	9.18204	0.51122	
		COST	248005	236,212	241,636	3506.311	
	III	$SAIFI_1^C$	5.8436	4.2788	4.9986	0.465248	0.74138
		$SAIDI_2^C$	7.8765	5.0667	6.35920	0.835414	
		COST	222836	185,337	202,586.54	11149.25	
	IV	$SAIFI_1^C$	8.2159	6.0158	7.0278	0.654137	0.565436
		$SAIDI_2^C$	8.3351	6.6482	7.42417	0.501551	
		COST	238127	198,055	216,488	11914.26	
	V	$SAIFI_1^C$	5.3539	3.9202	4.20694	0.758643	0.363858
		$SAIDI_2^C$	6.8445	4.4028	5.62365	1.22085	
		COST	175,675	146,112	157,937	14879.72	

Wilcoxon Rank-Sum Test for Reliability Optimization in Power Distribution Systems (PDS)

In this chapter we have performed the Wilcoxon Rank-Sum statistical test to compare the significance of population1 over population2.Here we have referred population1 for population generated by algorithm using the proposed model of Tippachon and Rerkpreedapong [240] and population2 for our proposed model.

Table 7.16: Ranks of objective-function value of different populations

Objective Function value			Ranks of Object Function value		Objective Function value			Ranks of Object Function value	
Run	population 1	population 2	for Population 1	for Population 2	Run	Population 1	Population2	for Population 1	for Population 2
1	9.0763	4.0784	88.5	24	26	7.8988	3.8343	79	17
2	8.916	7.1402	84.5	71	27	4.2788	4.2788	28.5	28.5
3	6.5656	6.5656	62.5	62.5	28	5.0667	5.0667	38.5	38.5
4	5.6093	5.6093	41.5	41.5	29	6.0158	3.8343	48	17
5	5.0347	5.0347	35.5	35.5	30	6.6482	3.2597	65	4
6	6.3747	6.3747	50.5	50.5	31	3.9202	3.9202	21.5	21.5
7	5.8001	5.8001	45.5	45.5	32	4.4028	4.4028	30.5	30.5
8	3.8343	3.8343	17	17	33	4.0784	4.0784	24	24
9	3.2597	3.2597	4	4	34	7.1402	7.1402	71	71
10	6.7213	6.7213	67.5	67.5	35	6.5656	6.5656	62.5	62.5
11	9.0763	3.8343	88.5	17	36	5.6093	5.6093	41.5	41.5
12	8.916	3.2597	84.5	4	37	5.0347	5.0347	35.5	35.5
13	6.5501	6.5501	58.5	58.5	38	6.3747	6.3747	50.5	50.5
14	6.3898	6.3898	54.5	54.5	39	5.8001	5.8001	45.5	45.5
15	7.8988	7.8988	79	79	40	3.8343	3.8343	17	17
16	7.7385	7.7385	76	76	41	3.2597	3.2597	4	4
17	3.7855	3.7855	11.5	11.5	42	6.7213	6.7213	67.5	67.5
18	3.6252	3.6252	8.5	8.5	43	9.0763	9.0763	88.5	88.5
19	4.1593	4.1593	26.5	26.5	44	8.916	8.916	84.5	84.5
20	4.8454	4.8454	32.5	32.5	45	6.5501	6.5501	58.5	58.5
21	7.5198	7.5198	73.5	73.5	46	6.3898	6.3898	54.5	54.5
22	7.5198	7.5198	73.5	73.5	47	7.8988	3.8343	79	17
23	8.4511	8.4511	81.5	81.5	48	7.7385	3.2597	76	4
24	8.4511	8.4511	81.5	81.5	49	3.7855	3.7855	11.5	11.5
25	8.916	3.2597	84.5	4	50	5.0347	5.0347	35.5	35.5

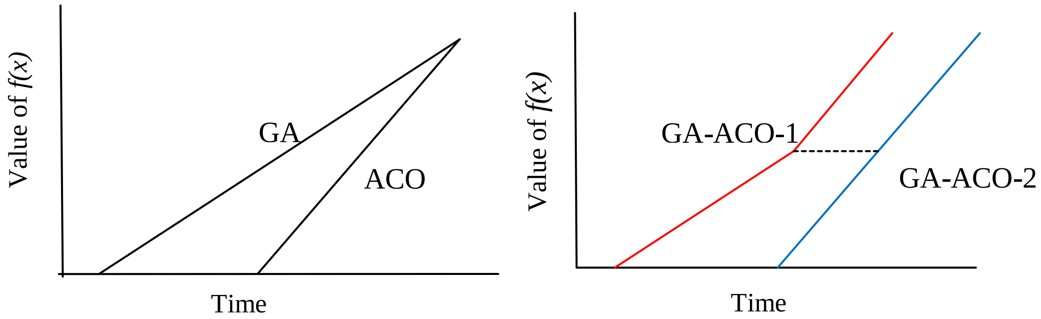
Table 7.17: p-value calculation of different populations

	Population 1	Population 2
count	50	50
Rank-sum	2275.5	1819.5
α	0.05	0.05
W’	2275.5	NA
W"	NA	1819.5
Mean	6.116746667	5.410406667
Variance	3.197052068	2.823868869
Standard deviation	1.752313086	1.680437107
p-value	1.8538	0

Conclusion: Since p-value of population2 (i.e., 0) is less than α value (i.e., 0.05) whereas p-value of population2 (i.e., 1.8538) is greater than α value (i.e., 0.05), therefore the population 2 has significance over population1 in the generated sample (population).

7.5.6 Discussion

This section discusses why our algorithm works better than others in this problem. We justify the novel operators and the way in which GA and ACO are combined to achieve our objective. Power crossover operation produces two chromosomes in each of which the value of gene gets reduced. In fact, this value is the intermediate value between the corresponding genes of the parent chromosomes. Depending on the choice of lambda, this value tilts towards one of the parent chromosome. The best and next best chromosomes are taken as the offspring. Non-uniform mutation used in this chapter searches uniformly in the initial stages and locally in the later stages. In initial stages, there is a small chance that the chromosomes will jump to its least or highest possible values. Therefore, this mutation operator along with the power crossover operator helps in searching the overall space and thus, the probability of convergence towards the global optimum increases. When GA is combined with ACO, it has the advantage of GA (which are its ability to compute feasible solutions and to avoid premature convergence) and the advantage of ACO (which is its ability to search fast over a subspace to local optima). The advantage of the combination is that ACO can come out of local optima if it is fed with a fresh set of feasible solution space obtained through GA. One generation of population of GA at the output can be fed to the input generation of a population to ACO in several ways. Lee et al. [341] used a straight forward approach to feed GA population to ACO population and vice versa. In our work, both GA and ACO are given fair opportunity to run their schemes. So, the best 50% of the last generation of GA is mixed with the best 50% of last generated solution by ACO and this mixture is fed to the ACO. This helps in faster convergence to a global optimum.



(a) The time taken by GA and ACO individually (b) Combining GA and ACO in two ways.

Figure 7.8:

Consider an extreme case with the following assumptions: (i) The local space where ACO searches contains the global optima of some function with just one variable. (ii) The output of GA is fed to the input of ACO only once. Figure 7.8 (a) depicts how GA and ACO converge to the optimum. The slope of GA is less than that of ACO. Figure 7.8 (b) depicts how GA-ACO would work if the solution obtained by GA is directly fed to ACO (See the line coloured red and labeled as GA-ACO-1). The same figure also shows that the GA-ACO works better if 50% of best solution of GA is mixed with 50% of the best already obtained by ACO (See the line coloured blue and labeled as GA-ACO-2). It can be seen that the curve GA-ACO-2 takes less time than GA-ACO-1

7.6 Conclusion

This chapter aims to demonstrating the superiority of a hybrid algorithm to optimize reliability indices and the total cost of switching and protection equipment installed in power distribution systems. GA-ACO hybrid algorithm has been chosen as the optimization method. It is shown that individually GA and ACO algorithms perform worse to the proposed hybrid algorithm. In the course of hybridization, novel operators have been used in both GA and ACO algorithms. The results prove that the performance of the proposed algorithm is better than an existing algorithm in the same domain, which may encourage other researchers in the field of electrical or power engineering optimization to use hybrid meta-heuristic algorithms. It is also shown that the proposed hybrid algorithm performs better than another GA-ACO hybrid algorithm and a GA-PSO algorithm. The future work could be to consider many other devices such as distributed generator (DG) and study the problem of optimizing reliability indices and the cost in power distribution networks. The value of load is uncertain in nature and it can be realistic if uncertainty can be applied to the models through fuzzy logic, which may lead to another direction of future research in this field.

CHAPTER 8

General Conclusion and Scope of Future Research

8.1 General Conclusion of the thesis

In this thesis, we have investigated different types of reliability optimization problems that have been formulated and solved as single and multi-objective constrained optimization problems with integer and/or mixed-integer variables. In chapter 2, the discussion is mainly focused on a literature survey on different evolutionary algorithms and reliability optimization problems. In chapter 3, we have discussed different research methodologies and mathematical backgrounds which are most essential to handle our proposed problems. In Chapter 4, GA-PSO algorithm for mixed-integer nonlinear programming problem in reliability optimization has been solved. In Chapter 5, multi-objective reliability-redundancy allocation problem by hybrid GA-PSO algorithm has been solved and computed results have been presented. Reliability Application of reliability redundancy allocation problem using hybrid GA-PSO in wireless sensor network (WSN) has been presented and discussed in chapter 6. We have also examined the reliability optimization in power distribution systems (PDS) using hybrid GA-ACO algorithm. This is addressed in Chapter 7.

In the whole work of the thesis, the reliability of each component/parameters are considered as either fixed value or fuzzy valued number depending upon the choices of the formulated problem. As a result the objective function as well as constraint of the formulated problems will be fixed valued/fuzzy valued, and these problems need to be optimized.

8.2 Scope of Further Research

For further researches, a large scope may arise from this thesis.

(i) The proposed techniques may be applied for solving real-life decision-making problems in the form of interval valued constrained optimization problems, interval valued multi-objective optimization problems, in different fields of engineering, management, manufacturing firms, etc.

(ii) In this thesis, we have solved all the optimization problems with the help of GA/PSO/DE/ACO as well as hybrid algorithms with precise/imprecise valued fitness functions in different engineering applications. These problems can be solved by other modified evolutionary algorithms/ hybrid algorithms.

(iii) In this thesis, we have used Big-M penalty techniques [292] to solve the constrained optimization problems. Alternatively, one may solve the same problem by using other penalty techniques.

(v) In Chapter 6, we have formulated and solved single-objective optimization problems considering only one objective, viz. system reliability for decision making in assessment of RRAP in WSN. One may extend the problem for multi-objectives, viz. system reliability, cost, volume, weight and cost. The same methodologies can be applied to solve multi-objective problems in the areas such as manufacturing, scheduling, marketing, assignment, transportation, inventory, etc.

(vi) In interval, fuzzy as well as stochastic environments, there are a lot of scopes to work in the area of multi-objective optimization problems.

(vii) There are a lot of scopes to work on the scalability and proof of correctness as well as convergence test of the proposed algorithms.

Bibliography

[1] Spall, James C. *Introduction to stochastic search and optimization: estimation, simulation, and control*. Vol. 65. John Wiley & Sons, 2005.

[2] Eiben, Agoston E., and James E. Smith. *Introduction to evolutionary computing*. Vol. 53. Heidelberg: springer, 2003.

[3] Prosser, Patrick. "Hybrid algorithms for the constraint satisfaction problem." *Computational intelligence* 9.3 (1993): 268-299.

[4] Mine, Hisashi. "Reliability of physical system." *Information Theory, IRE Transactions on* 5.5 (1959): 138-151.

[5] Pandey, Ajeet Kumar, and N. Goyal. *Early Software Reliability Prediction*. Springer, India, 2015.

[6] Avizienis, Algirdas. "Fault-tolerant systems." IEEE Trans. Computers 25.12 (1976): 1304-1312..

[7] Fiacco, Anthony V., and Garth P. McCormick. *Nonlinear programming: sequential unconstrained minimization techniques*. Society for Industrial and Applied Mathematics, 1990.

[8] Rausand, Marvin, and H. Ã. Arnljot. *System reliability theory: models, statistical methods, and applications*. Vol. 396. John Wiley & Sons, 2004.

[9] Kuo, Way. *Optimal reliability design: fundamentals and applications*. Cambridge university press, 2001.

[10] Massim, Y., et al. "Optimal design and reliability evaluation of multi-state series-parallel power systems." *Nonlinear Dynamics* 40.4 (2005): 309-321.

[11] Martz, H. F., and R. A. Wailer. "Bayesian reliability analysis of complex series/parallel systems of binomial subsystems and components." Technometrics 32.4 (1990): 407-416.

[12] Sahner, Robin A., and Kishor S. Trivedi. "Reliability modeling using SHARPE." *Reliability, IEEE Transactions on* 36.2 (1987): 186-193.

[13] Chiang, Dalen T., and Shun-Chen Niu. "Reliability of consecutive-k-out-of-n: F system." *IEEE Transactions on Reliability* 1 (1981): 87-89.

[14] Bremermann, Hans J. "Optimization through evolution and recombination." *Self-organizing systems* 93 (1962): 106.

[15] Friedberg, Richard M. "A learning machine: Part I." *IBM Journal of Research and Development* 2.1 (1958): 2-13.

[16] Friedberg, Richard M., Dunham B., and North J. H. "A learning machine:Part II," *IBM Journal of Research and Development* 3.7 (1959): 282–287.

[17] Box, George EP. "Evolutionary operation: A method for increasing industrial productivity." *Applied statistics* (1957): 81-101.

[18] Fogel, David B. *Evolutionary computation: toward a new philosophy of machine intelligence*. Vol. 1. John Wiley & Sons, 2006.

[19] Holland, John H. "Outline for a logical theory of adaptive systems." *Journal of the ACM (JACM)* 9.3 (1962): 297-314.

[20] Rechenberg, Ingo. "Cybernetic solution path of an experimental problem." (1965).

[21] Schwefel, H. P. *Projekt MHD-Staustrahlrohr: Experimentelle Optimierung einer Zweiphasendüse, Teil I*. Technischer Bericht 11.034/68, 35, AEG Forschungsinstitut, Berlin, Germany, 1968.

[22] Fogel, Lawrence J. "Autonomous automata." *Industrial research* 4 (1962): 14-19.

[23] Alander, Jarmo. T. "Indexed bibliography of genetic algorithms papers of 1996," University of Vaasa, Department of Information Technology and Production Economics, Rep. 94-1-96, 1995, (ftp.uwasa.fi, cs/report94-1,ga96bib.ps.Z).

[24] Holland, John Henry. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[25] Holland, John H., and Judith S. Reitman. "Cognitive systems based on adaptive algorithms." *Acm Sigart Bulletin* 63 (1977): 49-49.

[26] De Jong, Kenneth Alan. "Analysis of the behavior of a class of genetic adaptive systems." (1975).

[27] De Jong, Kenneth A. "On Using Genetic Algorithms to Search Program Spaces." *ICGA*. 1987.

[28] De Jong, Kenneth A. "Are genetic algorithms function optimizers?." *PPSN*. Vol. 2. 1992.

[29] De Jong, Kenneth A. "Genetic algorithms are NOT function optimizers." *Foundations of genetic algorithms* 2 (1993): 5-17.

[30] Goldberg, David E. "Genetic algorithms and rule learning in dynamic system control." *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*. 1985.

[31] Goldberg, David E. "Genetic algorithms in search optimization and machine learning." (1989).

[32] Goldberg, David E. "The theory of virtual alphabets." *International Conference on Parallel Problem Solving from Nature*. Springer, Berlin, Heidelberg, 1990.

[33] Goldberg, David E., Kalyanmoy Deb, and James H. Clark. "Genetic algorithms, noise, and the sizing of populations." *Urbana* 51 (1991): 61801.

[34] Goldberg, David E., et al. "RapidAccurate Optimization of Difficult Problems Using Fast Messy Genetic Algorithms." *ICGA*. Vol. 5. 1993.

[35] Davis, Lawrence. "Handbook of genetic algorithms." (1991).

[36] Eshelman, Larry J., and J. David Schaffer. "Crossover's niche." *Proceedings of the 5th International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc., 1993.

[37] Eshelman, Larry J., and J. David Schaffer. "Productive recombination and propagating and preserving schemata." *Foundations of genetic algorithms* 3 (2014): 299-313.

[38] Forrest, Stephanie, and Melanie Mitchell. "What makes a problem hard for a genetic algorithm? Some anomalous results and their explanation." *Machine Learning* 13.2-3 (1993): 285-319.

[39] Grefenstette, John J. "Optimization of control parameters for genetic algorithms." *IEEE Transactions on systems, man, and cybernetics* 16.1 (1986): 122-128.

[40] Grefenstette, John J. "Incorporating problem specific knowledge in genetic algorithms." *Genetic algorithms and simulated annealing* (1987): 42-60.

[41] Grefenstette, John J. "Conditions for implicit parallelism." *Rawlins* 144 (1995): 252-264.

[42] Grefenstette, John J. "Deception Considered Harmful sk." *Foundations of Genetic Algorithms 1993 (FOGA 2)* 2 (2014): 75.

[43] Koza, John R. "Hierarchical Genetic Algorithms Operating on Populations of Computer Programs." *IJCAI*. 1989.

[44] Koza, John R. *Genetic programming: on the programming of computers by means of natural selection*. Vol. 1. MIT press, 1992.

[45] Mitchell, Melanie. *An introduction to genetic algorithms*. MIT press, 1998.

[46] Riolo, Rick L. "The emergence of coupled sequences of classifiers." *Proceedings of the 3rd International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc., 1989.

[47] Riolo, Rick L. "The emergence of default hierarchies in learning classifier systems." *Proceedings of the 3rd International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc., 1989.

[48] Schaffer, J. David. "Multiple objective optimization with vector evaluated genetic algorithm." *Proceeding of the First International Conference of Genetic Algorithms and Their Application*. 1985.

[49] Schaffer, J. David, and Larry J. Eshelman. "On Crossover as an Evolutionarily Viable Strategy." *ICGA*. Vol. 91. 1991.

[50] Schaffer, J. David, and Amy Morishima. "An adaptive crossover distribution mechanism for genetic algorithms." *Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc, 1987.

[51] Fogel, L. J. “On the organization of intellect,” Ph.D. dissertation, University of California, Los Angeles, 1964.

[52] Burgin, G. "On playing two-person zero-sum games against nonminimax players." *IEEE Transactions on Systems Science and Cybernetics* 5.4 (1969): 369-370.

[53] Burgin, G. “Systems identification by quasilinearization and evolutionary programming,” J. Cybern., vol. 3, no. 2, pp. 56–75, 1973.

[54] Atmar III, Joe Wirt. "Speculation on the evolution of intelligence and its possible realization in machine form." (1976).

[55] Fogel, Lawrence J., Alvin J. Owens, and Michael J. Walsh. "Artificial intelligence through simulated evolution." (1966).

[56] Fogel, David B. "An evolutionary approach to the traveling salesman problem." *Biological Cybernetics* 60.2 (1988): 139-144.

[57] Fogel, David Bruce. "Evolving artificial intelligence." (1992).

[58] Rechenberg, Ingo. “Evolutionsstrategie Optimierung technischer Systeme nach Prinzipien der biologishen Evolution." *mit einem Nachwort von Manfred Eigen, Friedrich Frommann Verlag, Struttgart-Bad Cannstatt* 45 (1973): 46-47.

[59] Rechenberg, Ingo. "Evolutionsstrategie’94, volume 1 of Werkstatt Bionik und Evolutionstechnik." *Frommann {Holzboog, Stuttgart* (1994).

[60] Schwefel, Hans-Paul. *Evolutionsstrategie und numerische Optimierung*. Diss. Technische Universität Berlin, 1975.

[61] Schwefel, Hans-Paul Paul. *Evolution and optimum seeking: the sixth generation*. John Wiley & Sons, Inc., 1993.

[62] Herdy, Michael. "Reproductive Isolation as Strategy Parameter in Hierarichally Organized Evolution Strategies." *PPSN*. 1992.

[63] Kursawe, Frank. "A variant of evolution strategies for vector optimization." *International Conference on Parallel Problem Solving from Nature*. Springer, Berlin, Heidelberg, 1990.

[64] Ostermeier, Andreas. "An Evolution Strategy with Momentum Adaptation of the Random Number Distribution." *PPSN*. 1992.

[65] Ostermeier, Andreas, Andreas Gawelczyk, and Nikolaus Hansen. "Step-size adaptation based on non-local use of selection information." *Parallel Problem Solving from Nature—PPSN III* (1994): 189-198.

[66 Rudolph, Günter. "Global optimization by means of distributed evolution strategies." *International Conference on Parallel Problem Solving from Nature*. Springer, Berlin, Heidelberg, 1990.

[67] Schwefel, Hans-Paul, and Günter Rudolph. "Contemporary evolution strategies." *European conference on artificial life*. Springer, Berlin, Heidelberg, 1995.

[68] Klockgether, Jurgen, and Hans-Paul Schwefel. "Two-phase nozzle and hollow core jet experiments." *Proc. 11th Symp. Engineering Aspects of Magnetohydrodynamics*. Pasadena, CA: California Institute of Technology, 1970.

[69] Grefenstette, John J., ed. Proc. 1st Int. Conf. on Genetic Algorithms and Their Applications. Hillsdale, NJ: Lawrence Erlbaum, 1985.

[70] Grefenstette, John J., ed. Proc. 2nd Int. Conf. on Genetic Algorithms and Their Applications. Hillsdale, NJ: Lawrence Erlbaum, 1987.

[71] Schaffer, J. D., ed., Proc. 3rd Int. Conf. on Genetic Algorithms. San Mateo, CA, Morgan Kaufmann, 1989.

[72] Belew, R. K., and Booker, L. B., eds., Proc. 4th Int. Conf. on Genetic Algorithms. San Mateo, CA, Morgan Kaufmann, 1991.

[73] Forrest, S., Ed., Proc. 5th Int. Conf. on Genetic Algorithms. San Mateo, CA: Morgan Kaufmann, 1993.

[74] Eshelman, L., Ed., Genetic Algorithms: Proc. 6th Int. Conf. San Francisco, CA: Morgan Kaufmann, 1995.

[75] Fogel, D.B., and Atmar, W., eds., Proc 1st Annu. Conf. on Evolutionary Programming. San Diego, CA: Evolutionary Programming Society, 1992.

[76] Fogel, D.B., and Atmar, W., eds., Proc. 2nd Annu. Conf. on Evolutionary Programming. San Diego, CA: Evolutionary Programming Society, 1993.

[77] Sebald A. V., and Fogel, L. J., eds., Proc. 3rd Annual Conf. on Evolutionary Programming. Singapore: World Scientific, 1994.

[78] McDonnell, J. R., Reynolds, R. G., and Fogel, D. B., eds., Proc. 4th Annu. Conf. on Evolutionary Programming. Cambridge, MA: MIT Press, 1995.

[79] Fogel, L. J., Angeline, P. J., and B¨ack, T., eds., Proc. 5th Annu. Conf. On Evolutionary Programming. Cambridge, MA: The MIT Press, 1996.

[80] Rawlins, G. J. E., ed., Foundations of Genetic Algorithms. San Mateo, CA: Morgan Kaufmann, 1991.

[81] Whitley, L. D., ed., Foundations of Genetic Algorithms 2. San Mateo, CA: Morgan Kaufmann, 1993.

[82] Vose M. D. and Whitley, L. D., eds., Foundations of Genetic Algorithms 3. San Francisco, CA: Morgan Kaufmann, 1995.

[83] Koza, J. R., Goldberg, D. E., Fogel, D. B. Riolo, R. L., eds., Genetic Programming 1996. Proc. 1st Annu. Conf. Cambridge, MA: MIT Press, 1996.

[84] Schwefel, H.P., and M¨anner, R., eds., Parallel Problem Solving from Nature—Proc. 1st Workshop PPSN I. Berlin, Germany: Springer, 1991, vol. 496 of Lecture Notes in Computer Science.

[85] M¨anner, R., Manderick, B., eds., Parallel Problem Solving from Nature 2. Amsterdam, The Netherlands: Elsevier, 1992.

[86] Davidor, Y., Schwefel, H.P., M¨anner, R., eds., Parallel Problem Solving from Nature—PPSN III, Int. Conf. on Evolutionary Computation. (Lecture Notes in Computer Science, vol. 866) Berlin: Springer, 1994.

[87] Voigt, H.M., Ebeling, W., Rechenberg, I., Schwefel, H.P., eds., Parallel Problem Solving from Nature IV. Proc. Int. Conf. on Evolutionary Computation. Berlin, Germany: Springer, 1996, vol. 1141 of Lecture Notes in Computer Science.

[88] Proc. 1st IEEE Conf. on Evolutionary Computation, Orlando, FL. Piscataway, NJ: IEEE Press, 1994.

[89] Proc. 2nd IEEE Conf. on Evolutionary Computation, Perth, Australia. Piscataway, NJ: IEEE Press, 1995.

[90] Proc. 3rd IEEE Conf. on Evolutionary Computation, Nagoya, Japan. Piscataway, NJ: IEEE Press, 1996.

[91] Proc. 4th IEEE Conf. on Evolutionary Computation, Indianapolis, IN. Piscataway, NJ: IEEE Press, 1997.

[92] Back, T., Evolutionary Algorithms in Theory and Practice. New York: Oxford Univ. Press, 1996.

[93] Back, T., Fogel, D. B., Michalewicz, Z., eds., Handbook of Evolutionary Computation. New York: Oxford Univ. Press and Institute of Physics, 1997.

[94] Kinnear, K. E., ed., Advances in Genetic Programming. Cambridge, MA: MIT Press, 1994.

[95] Michalewicz, Zbigniew. *Genetic algorithms+ data structures= evolution programs*. Springer Science & Business Media, 2013.

[96] Fogel, David B. *Evolutionary computation: the fossil record*. Wiley-IEEE Press, 1998.

[97] Branke, Jürgen. "Memory enhanced evolutionary algorithms for changing optimization problems." *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*. Vol. 3. IEEE, 1999.

[98] Bäck, Thomas, David B. Fogel, and Zbigniew Michalewicz, eds. *Evolutionary computation 1: Basic algorithms and operators*. Vol. 1. CRC press, 2000.

[99] Sinha, Nidul, R. Chakrabarti, and P. K. Chattopadhyay. "Evolutionary programming techniques for economic load dispatch." *IEEE Transactions on evolutionary computation* 7.1 (2003): 83-94.

[100 Jin, Yaochu. "A comprehensive survey of fitness approximation in evolutionary computation." *Soft Computing-A Fusion of Foundations, Methodologies and Applications* 9.1 (2005): 3-12.

[101] De Jong, Kenneth A. *Evolutionary computation: a unified approach*. MIT press, 2006.

[102] Lim, Dudy, et al. "Generalizing surrogate-assisted evolutionary computation." *IEEE Transactions on Evolutionary Computation* 14.3 (2010): 329-355.

[103] Jin, Yaochu. "Surrogate-assisted evolutionary computation: Recent advances and future challenges." *Swarm and Evolutionary Computation* 1.2 (2011): 61-70.

[104] Bäck, Thomas, and Hans-Paul Schwefel. "An overview of evolutionary algorithms for parameter optimization." *Evolutionary computation* 1.1 (1993): 1-23.

[105] Dasgupta, Dipankar, and Zbigniew Michalewicz, eds. *Evolutionary algorithms in engineering applications*. Springer Science & Business Media, 2013.

[106] Holland, John H. "Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence." *Ann Arbor, MI: University of Michigan Press* (1975): 439-444.

[107] Schaffer, James David. "Some experiments in machine learning using vector evaluated genetic algorithms (artificial intelligence, optimization, adaptation, pattern recognition)." (1984).

[108] Michalewicz, Zbigniew, and Marc Schoenauer. "Evolutionary algorithms for constrained parameter optimization problems." *Evolutionary computation* 4.1 (1996): 1-32.

[109] Koza, John R. "Genetic programming as a means for programming computers by natural selection." *Statistics and computing* 4.2 (1994): 87-112.

[110] Liu, Baoding, and Boading Liu. *Theory and practice of uncertain programming*. Vol. 239. Berlin: Springer, 2009.

[111] Bielli, Maurizio, and Pasquale Carotenuto. "Genetic Algorithms and Transportation Analysis: Review and Perspectives for Bus Network Optimization." *New Analytical Advances in Transportation and Spatial Dynamics*. Routledge, 2018. 35-48.

[112] Holland, John H. "The global economy as an adaptive process." *The economy as an evolving complex system*. CRC Press, 2018. 117-124.

[113] Anderson, Philip W. *The economy as an evolving complex system*. CRC Press, 2018.

[114] Kadri, Roubila Lilia, and Fayez F. Boctor. "An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case." *European Journal of Operational Research* 265.2 (2018): 454-462.

[115] Corus, Dogan, and Per Kristian Lehre. "Theory Driven Design of Efficient Genetic Algorithms for a Classical Graph Problem." *Recent Developments in Metaheuristics*. Springer, Cham, 2018. 125-140.

[116] Kim, Heungseob, and Pansoo Kim. "Reliability–redundancy allocation problem considering optimal redundancy strategy using parallel genetic algorithm." *Reliability Engineering & System Safety* 159 (2017): 153-160.

[117] Biswas, Swagata, Ria Das, and Punyasha Chatterjee. "Energy-Efficient Connected Target Coverage in Multi-hop Wireless Sensor Networks." *Industry Interactive Innovations in Science, Engineering and Technology*. Springer, Singapore, 2018. 411-421.

[118] Abdul, Azeem Mohammed, et al. "Analysis of Genetic Algorithm for Effective Power Delivery and with Best Upsurge." *Data Engineering and Intelligent Computing*. Springer, Singapore, 2018. 1-9.

[119] Eberhart, Russell, and James Kennedy. "A new optimizer using particle swarm theory." *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*. IEEE, 1995.

[120] Parsopoulos, K. E., et al. "Stretching technique for obtaining global minimizers through particle swarm optimization." *Proceedings of the particle swarm optimization Workshop*. Vol. 29. Indianapolis, USA, 2001.

[121] Parsopoulos, K. E., and M. N. Vrahatis. "Particle swarm optimizer in noisy and continuously changing environment." *methods* 5.6 (2001): 23.

[122] Fourie, P. C., and Albert A. Groenwold. "Particle swarms in size and shape optimization." *Proceedings of the international workshop on multidisciplinary design optimization, Pretoria, South Africa*. 2000.

[123] Fourie, P. C., and Albert A. Groenwold. "Particle swarms in topology optimization." *Proceedings of the fourth world congress of structural and multidisciplinary optimization*. 2001.

[124] Rinsch, Christopher L., et al. "Pulsed radio frequency plasma polymerization of allyl alcohol: controlled deposition of surface hydroxyl groups." Langmuir 12.12 (1996): 2995-3002.

[125] Kennedy, James. "Particle swarm optimization." Encyclopedia of Machine Learning.Springer US, 2010.760-766.

[126]Kennedy, James, et al. Swarm intelligence. Morgan Kaufmann, 2001.

[127]Shi, Yuhui, and Russell Eberhart. "A modified particle swarm optimizer." Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on. IEEE, 1998.

[128] Shi, Yuhui, and Russell C. Eberhart. "Parameter selection in particle swarm optimization." Evolutionary programming VII. Springer Berlin Heidelberg, 1998.

[129] Clerc, Maurice. "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization."Evolutionary Computation, 1999.CEC 99.Proceedings of the 1999 Congress on.Vol. 3.IEEE, 1999.

[130] Eberhart, Russell C., and Yuhui Shi. "Comparison between genetic algorithms and particle swarm optimization."Evolutionary Programming VII.Springer Berlin Heidelberg, 1998.

[131] Engelbrecht, Andries. "Particle swarm optimization." Proceedings of the 2014 conference companion on Genetic and evolutionary computation companion. ACM, 2014.

[132] Price, K. "Differential evolution vs. the functions of the 2nd ICEC." Proceeding of 1997 IEEE International Conference on Evolutionary Computation. 1997.

[133] Price, Kenneth, and Rainer Storn. "Differential evolution: A simple evolution strategy for fast optimization." Dr. Dobb’s journal 22.4 (1997): 18-24.

[134] Storn, Rainer, and Kenneth Price. "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces." Journal of global optimization 11.4 (1997): 341-359.

[135] Storn, Rainer, and Kenneth Price. "Minimizing the real functions of the ICEC'96 contest by differential evolution." Evolutionary Computation, 1996., Proceedings of IEEE International Conference on. IEEE, 1996.

[136] Storn, Rainer. "On the usage of differential evolution for function optimization." *Fuzzy Information Processing Society, 1996. NAFIPS., 1996 Biennial Conference of the North American*. IEEE, 1996.

[137] Brest, Janez, et al. "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems." IEEE transactions on evolutionary computation 10.6 (2006): 646-657.

[138] Qin, A. Kai, and Ponnuthurai N. Suganthan. "Self-adaptive differential evolution algorithm for numerical optimization." *Evolutionary Computation, 2005. The 2005 IEEE Congress on*. Vol. 2. IEEE, 2005.

[139] Hansen, Nikolaus, and Andreas Ostermeier. "Completely derandomized self-adaptation in evolution strategies." *Evolutionary computation* 9.2 (2001): 159-195.

[140] Auger, Anne, and Nikolaus Hansen. "A restart CMA evolution strategy with increasing population size." *Evolutionary Computation, 2005. The 2005 IEEE Congress on*. Vol. 2. IEEE, 2005.

[141] Qin, A. Kai, Vicky Ling Huang, and Ponnuthurai N. Suganthan. "Differential evolution algorithm with strategy adaptation for global numerical optimization." *IEEE transactions on Evolutionary Computation* 13.2 (2009): 398-417.

[142] Brest, Janez, and Mirjam Sepesy Maučec. "Population size reduction for the differential evolution algorithm." *Applied Intelligence* 29.3 (2008): 228-247.

[143] Rahnamayan, Shahryar, Hamid R. Tizhoosh, and Magdy MA Salama. "Opposition-based differential evolution." *IEEE Transactions on Evolutionary computation* 12.1 (2008): 64-79.

[144] Das, Swagatam, et al. "Differential evolution using a neighborhood-based mutation operator." *IEEE Transactions on Evolutionary Computation* 13.3 (2009): 526-553.

[145] Zhang, Jingqiao, and Arthur C. Sanderson. "JADE: adaptive differential evolution with optional external archive." *IEEE Transactions on evolutionary computation* 13.5 (2009): 945-958.

[146] Liang, Jane-Jing, Ponnuthurai Nagaratnam Suganthan, and Kalyanmoy Deb. "Novel composition test functions for numerical global optimization." *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*. IEEE, 2005.

[147] Nelder, John A., and Roger Mead. "A simplex method for function minimization." *The computer journal* 7.4 (1965): 308-313.

[148] Price, W. L. "Global optimization by controlled random search." *Journal of Optimization Theory and Applications* 40.3 (1983): 333-348.

[149] Sarker, Ruhul, Saber M. Elsayed, and Tapabrata Ray. "Differential evolution with dynamic parameters selection for optimization problems." Evolutionary Computation, IEEE Transactions on 18.5 (2014): 689-707.

[150] Dorigo, Marco, and Gianni Di Caro. "Ant colony optimization: a new meta-heuristic." *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*. Vol. 2. IEEE, 1999.

[151] Dorigo, Marco, Gianni Di Caro, and Luca M. Gambardella. "Ant algorithms for discrete optimization." *Artificial life* 5.2 (1999): 137-172.

[152] Dorigo, Marco, and Thomas Stutzle. "The ant colony optimization metaheuristic: Algorithms, applications, and advances." *International series in operations research and management science* (2003): 251-286.

[153] Dorigo, Marco. "Optimization, learning and natural algorithms." *Ph. D. Thesis, Politecnico di Milano, Italy* (1992).

[154] Dorigo, Marco, Vittorio Maniezzo, and Alberto Colorni. "The ant system: An autocatalytic optimizing process." (1991).

[155] Dorigo, Marco, et al. "Positive feedback as a search strategy." (1991).

[156] Dorigo, Marco, Vittorio Maniezzo, and Alberto Colorni. "Ant system: optimization by a colony of cooperating agents." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 26.1 (1996): 29-41.

[157] Applegate, David L., et al. The traveling salesman problem: a computational study. Princeton university press, 2011.

[158] Gambardella, Luca Maria, and Marco Dorigo. "An ant colony system hybridized with a new local search for the sequential ordering problem." *INFORMS Journal on Computing* 12.3 (2000): 237-255.

[159] Blum, Christian. "Beam-ACO—Hybridizing ant colony optimization with beam search: An application to open shop scheduling." *Computers & Operations Research* 32.6 (2005): 1565-1591.

[160] Blum, Christian. "Beam-ACO for simple assembly line balancing." *INFORMS Journal on Computing* 20.4 (2008): 618-627.

[161] Blum, Christian. Theoretical and Practical Aspects of Ant Colony Optimization.PhD thesis, IRIDIA, Universit´eLibre de Bruxelles, Brussels, Belgium, 2004.

[162] Shmygelska, Alena, and Holger H. Hoos. "An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem." *BMC bioinformatics* 6.1 (2005): 30.

[163] Blum, Christian, Mateu Yábar Vallès, and Maria J. Blesa. "An ant colony optimization algorithm for DNA sequencing by hybridization." *Computers & Operations Research* 35.11 (2008): 3620-3635.

[164] Korb, Oliver, Thomas Stützle, and Thomas E. Exner. "An ant colony optimization approach to flexible protein–ligand docking." *Swarm Intelligence* 1.2 (2007): 115-134.

[165] Di Caro, Gianni, and Marco Dorigo. "AntNet: Distributed stigmergetic control for communications networks." *Journal of Artificial Intelligence Research* 9 (1998): 317-365.

[166] Liao, Tianjun, et al. "Ant colony optimization for mixed-variable optimization problems." Evolutionary Computation, IEEE Transactions on 18.4 (2014): 503-518.

[167] Mohan, B. Chandra, and R. Baskaran. "A survey: Ant Colony Optimization based recent research and implementation on several engineering domain." Expert Systems with Applications 39.4 (2012): 4618-4627.

[168] Allella, Flavio, Elio Chiodo, and Davide Lauria. "Optimal reliability allocation under uncertain conditions, with application to hybrid electric vehicle design." International Journal of Quality & Reliability Management 22.6 (2005): 626-641.

[169] Yalaoui, Alice, Eric Châtelet, and Chengbin Chu. "A new dynamic programming method for reliability & redundancy allocation in a parallel-series system." *IEEE transactions on reliability* 54.2 (2005): 254-261.

[170] Salazar, Daniel, Claudio M. Rocco, and Blas J. Galván. "Optimization of constrained multiple-objective reliability problems using evolutionary algorithms." *Reliability Engineering & System Safety* 91.9 (2006): 1057-1070.

[171] Kim, Jae-Hwan, and Bong-Jin Yum. "A heuristic method for solving redundancy optimization problems in complex systems." *IEEE Transactions on Reliability* 42.4 (1993): 572-578.

[172] Coit, David W., and Alice E. Smith. "Reliability optimization of series-parallel systems using a genetic algorithm." *IEEE Transactions on reliability* 45.2 (1996): 254-260.

[173] Coit, David W., and Alice E. Smith. "Redundancy allocation to maximize a lower percentile of the system time-to-failure distribution." *IEEE Transactions on Reliability* 47.1 (1998): 79-87.

[174] Kuo, Way, and V. Rajendra Prasad. "An annotated overview of system-reliability optimization." *IEEE Transactions on reliability* 49.2 (2000): 176-187.

[175] Liang, Yun-Chia, and Alice E. Smith. "An ant colony optimization algorithm for the redundancy allocation problem (RAP)." *IEEE Transactions on reliability* 53.3 (2004): 417-423.

[176] Ramirez-Marquez, Jose E., and David W. Coit. "A heuristic for solving the redundancy allocation problem for multi-state series-parallel systems." Reliability Engineering & System Safety 83.3 (2004): 341-349.

[177] Yun, Won Young, and Jong Woon Kim. "Multi-level redundancy optimization in series systems." *Computers & Industrial Engineering* 46.2 (2004): 337-346.

[178] Nourelfath, Mustapha, and Nabil Nahas. "Artificial neural networks for reliability maximization under budget and weight constraints." *Journal of Quality in Maintenance Engineering* 11.2 (2005): 139-151.

[179] You, Peng-Sheng, and Ta-Cheng Chen. "An efficient heuristic for series–parallel redundant reliability problems." *Computers & operations research* 32.8 (2005): 2117-2127.

[180] Agarwal, Manju, and Rashika Gupta. "Genetic search for redundancy optimization in complex systems." *Journal of Quality in maintenance Engineering* 12.4 (2006): 338-353.

[181] Coit, D. W. and Konak, A. (2006), Multiple weighted objectives heuristic for the redundancy allocation problem, IEEE Transactions on Reliability, 55, 551-558.

[182] Ha, Chunghun, and Way Kuo. "Reliability redundancy allocation: An improved realization for nonconvex nonlinear programming problems." *European Journal of Operational Research* 171.1 (2006): 24-38.

[183] Tian, Zhigang, and Ming J. Zuo. "Redundancy allocation for multi-state systems using physical programming and genetic algorithms." *Reliability Engineering & System Safety* 91.9 (2006): 1049-1056.

[184] Liang, Yun-Chia, and Yi-Ching Chen. "Redundancy allocation of series-parallel systems using a variable neighborhood search algorithm." *Reliability Engineering & System Safety* 92.3 (2007): 323-331.

[185] Nahas, Nabil, Mustapha Nourelfath, and Daoud Ait-Kadi. "Coupling ant colony and the degraded ceiling algorithm for the redundancy allocation problem of series–parallel systems." *Reliability Engineering & System Safety* 92.2 (2007): 211-222.

[186] Federowicz, A. J., and M. Mazumdar. "Use of geometric programming to maximize reliability achieved by redundancy." *Operations Research* 16.5 (1968): 948-954.

[187] Tillman, Frank A., Ching-Lai Hwang, and Way Kuo. "Optimization Techniques for System Reliability with Redundancy�A Review." *IEEE Transactions on Reliability* 26.3 (1977): 148-155.

[188] Misra, K. B., and Sharma U. "An efficient algorithm to solve integer-programming problems arising in system-reliability design." *IEEE Transactions on Reliability* 40.1 (1991): 81-91.
[189] Dhingra, Anoop K. "Optimal apportionment of reliability and redundancy in series systems under multiple objectives." *IEEE Transactions on reliability* 41.4 (1992): 576-582.

[190] Painton, Laura, and James Campbell. "Genetic algorithms in optimization of system reliability." *IEEE Transactions on reliability* 44.2 (1995): 172-178.

[191] Ha, Chunghun, and Way Kuo. "Multi-path approach for reliability-redundancy allocation using a scaling method." *Journal of Heuristics* 11.3 (2005): 201-217.

[192] Ha, Chunghun, and Way Kuo. "Multi-path heuristic for redundancy allocation: the tree heuristic." *IEEE Transactions on Reliability* 55.1 (2006): 37-43.

[193] Chen, Ta-Cheng. "IAs based approach for reliability redundancy allocation problems." *Applied Mathematics and Computation* 182.2 (2006): 1556-1567.

[194] Kim, Ho-Gyun, Chang-Ok Bae, and Dong-Jun Park. "Reliability-redundancy optimization using simulated annealing algorithms." *Journal of Quality in Maintenance Engineering* 12.4 (2006): 354-363.

[195] Zhao, Ruiqing, and Baoding Liu. "Stochastic programming models for general redundancy-optimization problems." *IEEE Transactions on Reliability* 52.2 (2003): 181-191.

[196] Zhao, Ruiqing, and Kaoping Song. "A hybrid intelligent algorithm for reliability optimization problems." *Fuzzy Systems, 2003. FUZZ'03. The 12th IEEE International Conference on*. Vol. 2. IEEE, 2003.

[197] Yu, Haiyang, et al. "Optimal design of a maintainable cold-standby system." *Reliability Engineering & System Safety* 92.1 (2007): 85-91.

[198] Prasad, V. Rajendra, and Way Kuo. "Reliability optimization of coherent systems." *IEEE Transactions on Reliability* 49.3 (2000): 323-330.

[199] Ramirez-Marquez, Jose Emmanuel, and David W. Coit. "Multi-state component criticality analysis for reliability improvement in multi-state systems." *Reliability Engineering & System Safety* 92.12 (2007): 1608-1619.

[200] Meziane, R., et al. "Reliability optimization using ant colony algorithm under performance and cost constraints." *Electric power systems research* 76.1-3 (2005): 1-8.

[201] Tian, Zhigang, Gregory Levitin, and Ming J. Zuo. "A joint reliability–redundancy optimization approach for multi-state series–parallel systems." *Reliability Engineering & System Safety* 94.10 (2009): 1568-1576.

[202] Li, Chun-yang, et al. "Heterogeneous redundancy optimization for multi-state series–parallel systems subject to common cause failures." *Reliability Engineering & System Safety* 95.3 (2010): 202-207.

[203] Caserta, Marco, and Stefan Voß. "An exact algorithm for the reliability redundancy allocation problem." European Journal of Operational Research 244.1 (2015): 110-116.

[204] Sahoo, Laxminarayan, et al. "An efficient GA–PSO approach for solving mixed- integer nonlinear programming problem in reliability optimization." Swarm and Evolutionary Computation 19 (2014): 43-51.

[205] Banerjee A. et al., A Novel Hybrid GA-PSO Optimization Technique: Application to Load Profiling Problems, Int. J. Energetica, Volume 63, ISSN: 1453-2360 , Nr. 4 / 2015 .pp 134-138.

[206] Banerjee A. et al., Reliability improvement and the importance of power consumption optimization in wireless sensor networks." Advanced Topics in Electrical Engineering (ATEE), 2015 9th International Symposium on. IEEE, 2015.

[207] Banerjee A. et al., A Fuzzy-ACO Algorithm to enhance Reliability Optimization through Energy Harvesting in WSN, International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) 3rd-5th March, 2016.

[208] Misra, Satyajayant, Martin Reisslein, and Guoliang Xue. "A survey of multimedia streaming in wireless sensor networks." Communications Surveys & Tutorials, IEEE 10.4 (2008): 18-39.

[209] Grossglauser M., and Tse, D., "Mobility increases the capacity of ad hoc wireless networks" In Proceedings of INFOCOM. Vol. 3, 2001, pp. 1360–1369.

[210] Diggavi, S., Grossglauser, M., and TSE, D., "Even one-dimensional mobility increases ad hoc wireless capacity", in Proceedings of ISIT. 352, 2002.

[211] Ailawadhi, V. , "Mobility issue in hybrid ad hoc wireless sensor networks". Ph.D. dissertation, Electrical Engineering Department, University of California, Los Angeles, Los Angeles, CA, 2002.

[212] Shah, R., Roy, S., Jain, S., and Brunette, W., "Data mules: Modeling a three-tier architecture for sparse sensor networks". In Proceedings of SNPA. pp. 30–41, 2003.

[213] Kansal, A., Rahimi, M., Kaiser, W.J. , Srivastava, M.B., Pottie, G.J., and Estrin, D., "Controlled mobility for sustainable wireless networks". In Proceedings of SECON, 2004.

[214] Tong, L., Zhao, Q., and Adireddy, S., "Sensor networks with mobile agents", In Proceedings of Military Communications Conference, pp. 688–693, 2003.

[215] Chakrabarti, A., Sabharwal, A., and Aazhang, B., "Communication Power Optimization in a Sensor Network with a Path-Constrained Mobile Observer". ACM Transactions on Sensor Networks, Vol. 2, No. 3, August 2006, pp. 297–324.

[216] Raghunathan, Vijay, et al. "Energy-aware wireless microsensor networks." Signal Processing Magazine, IEEE 19.2 (2002): 40-50.

[217] Akyildiz, Ian F., et al. "A survey on sensor networks." IEEE Communications magazine 40.8 (2002): 102-114.

[218] Kim, Sukun, Rodrigo Fonseca, and David Culler. "Reliable transfer on wireless sensor networks." Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on. IEEE, 2004.

[219]Tillman, F. A., Hwang, C. L., & Kuo, W. (1977). "Optimization Techniques for System Reliability with Redundancy-A Review". Reliability, IEEE Transactions on, 26(3), 148-155.

[220]Dima, S. M., Panagiotou, C., Tsitsipis, D., Antonopoulos, C., Gialelis, J., & Koubias, S. (2014). Performance evaluation of a WSN system for distributed event detection using fuzzy logic. Ad Hoc Networks, 23, 87-108.

[221]Cinque, M., Cotroneo, D., De Caro, G., & Pelella, M. (2006, June). Reliability requirements of wireless sensor networks for dynamic structural monitoring. In International Workshop on Applied Software Reliability (WASR 2006) (pp. 8-13).

[222] Shrestha, Akhilesh, and Liudong Xing. "Quantifying application communication reliability of wireless sensor networks." International Journal of Performability Engineering 4.1 (2008): 43-56.

[223] Shaikh, F. K., Khelil, A., & Suri, N. (2007, February). On modeling the reliability of data transport in wireless sensor networks. In Parallel, Distributed and Network-Based Processing, 2007. PDP'07. 15th EUROMICRO International Conference on (pp. 395-402). IEEE.

[224] Banerjee, Avishek, et al. "Decision making in assessment of RRAP of WSN using fuzzy-hybrid approach." 2015 IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS). IEEE, 2015.

[225] Parameswaran, Ambili Thottam, Mohammad Iftekhar Husain, and Shambhu Upadhyaya. "Is rssi a reliable parameter in sensor localization algorithms: An experimental study." Field Failure Data Analysis Workshop (F2DA09). 2009.

[226] Allan, Ronald N. Reliability evaluation of power systems. Springer Science & Business Media, 2013.

[227] Pregelj, Aleksandar, Miroslav Begovic, and Ajeet Rohatgi. "Recloser allocation for improved reliability of DG-enhanced distribution networks." IEEE Transactions on Power Systems 21.3 (2006): 1442-1449.

[228] Brown, Richard E. Electric power distribution reliability. CRC press, 2008. doi:10.1201/9780849375682

[229] IEEE 1366 , IEEE Guide for Electric Power Distribution Reliability Indices.(n.d.). doi:10.1109/ieeestd.2001.94438 (2012)

[230] Billinton, Roy, and Ronald N. Allan. "Distribution systems—basic techniques and radial networks." Reliability Evaluation of Power Systems. Springer US, 1984. 210-239.doi:10.1007/978-1-4899-1860-4_7

[231] Hilber, Patrik, and Lina Bertling. "A method for extracting reliability importance indices from reliability simulations of electrical networks." 15th Power Systems Computation Conference, PSCC 2005, Liege, Belgium, 22 August 2005 through 26 August 2005. Power Systems Computation Conference (PSCC), 2005.

[232] Tirapong, K., and S. Titti. "Reliability improvement of distribution system using Reliability Centered Maintenance." T&D Conference and Exposition, 2014 IEEE PES. IEEE, 2014. doi:10.1109/tdc.2014.6863360

[233] Li, Fangxing, and Richard E. Brown. "A cost-effective approach of prioritizing distribution maintenance based on system reliability." IEEE transactions on power delivery 19.1 (2004): 439-441.

[234] Ramírez-Rosado, Ignacio J., and José Antonio Domínguez-Navarro. "New multiobjective tabu search algorithm for fuzzy optimal planning of power distribution systems." IEEE Transactions on Power systems 21.1 (2006): 224-233.

[235] Teng, Jen-Hao, and Chan-Nan Lu. "Feeder-switch relocation for customer interruption cost minimization." IEEE transactions on power delivery 17.1 (2002): 254-259.

[236] Soudi, F., and K. Tomsovic. "Optimized distribution protection using binary programming." IEEE transactions on power delivery 13.1 (1998): 218-224.

[237] Teng, Jen-Hao, and Yi-Hwa Liu. "A novel ACS-based optimum switch relocation method." IEEE Transactions on power systems 18.1 (2003): 113-120.

[238] Da Silva, Luis G. Wesz, et al. "Optimised placement of control and protective devices in electric distribution systems through reactive tabu search algorithm." Electric Power Systems Research 78.3 (2008): 372-381.

[239] Popović DH, Greatbanks JA, Begović M, Pregelj A (2005) Placement of distributed generators and reclosers for distribution network security and reliability. Int J of Electrical Power & Energy Systems 27(5): 398-408.

[240] Tippachon, Wiwat, and Dulpichet Rerkpreedapong. "Multiobjective optimal placement of switches and protective devices in electric power distribution systems using ant colony optimization." Electric Power Systems Research 79.7 (2009): 1171-1178.

[241] Ray, Saheli, Aniruddha Bhattacharya, and Subhadeep Bhattacharjee. "Optimal placement of switches in a radial distribution network for reliability improvement." International Journal of Electrical Power & Energy Systems 76 (2016): 53-68.

[242] Kao, Yi-Tung, and Erwie Zahara. "A hybrid genetic algorithm and particle swarm optimization for multimodal functions." Applied soft computing 8.2 (2008): 849-857.

[243] Saravanan, C., and M. A. Panneerselvam. "A comprehensive analysis for extracting single diode PV model parameters by hybrid GA-PSO algorithm." International Journal of Computer Applications 78.8 (2013): 16-19.

[244] Jeong, Shinkyu, et al. "Development and investigation of efficient GA/PSO-hybrid algorithm applicable to real-world design optimization." IEEE Computational Intelligence Magazine 4.3 (2009): 36-44.

[245] Sheikhan, Mansour, and Najmeh Mohammadi. "Neural-based electricity load forecasting using hybrid of GA and ACO for feature selection." Neural Computing and Applications 21.8 (2012): 1961-1970.

[246] Zukhri, Zainudin, and Irving Vitra Paputungan. "A hybrid optimization algorithm based on genetic algorithm and ant colony optimization." International Journal of Artificial Intelligence & Applications 4.5 (2013): 63.

[247] Fidanova, Stefka, Marcin Paprzycki, and Olympia Roeva. "Hybrid GA-ACO algorithm for a model parameters identification problem." 2014 Federated Conference on Computer Science and Information Systems. IEEE, 2014.

[248] Tillman, F. A., Hwang, C. L. and Kuo, W., Optimization technique for system reliability with redundancy: A Review, IEEE Transactions on Reliability 26(1977):148-155.

[249] Tillman, F. A., Hwang, C. L. and Kuo, W., Determining Component Reliability and Redundancy for Optimum System Reliability, IEEE Transactions on Reliability, 26 (1977): 162-165.

[250] Tillman, F. A., Hwang, C. L. and Kuo, W. (1980), System Effectiveness Models: An Annotated Bibliography, IEEE Transactions on Reliability 29(1980): 295-304.

[251] Nakagawa, Y., Nakashima, K., and Hattori, Y., Optimal Reliability Allocation by Branch-and -Bounded technique, IEEE Transactions Reliability, 27(1978):31-38.

[252] Sharma, Usha, and K. B. Misra. "An Efficient Algorithm to Solve Integer-programming Problems in Reliability Optimisation." International Journal of Quality & Reliability Management 7.5 (1990).

[253] Chern, M. S., On the computational complexity of reliability redundancy allocation in a series system, Operations Research Letter, 11(1992): 309-315.

[254] Ohtagaki, H., Nakagawa, Y., Iwasaki, A. and Narihisa, H., Smart greedy procedure for solving a nonlinear knapsack class of reliability optimization problems, Math. Comput. Modeling, 22(1995): 261-272.

[255] Kuo, W., Prasad, V. R., Tillman, F. A. and Hwang, C. L., Optimal Reliability Design: Fundamentals and Application, Cambridge University Press (2001).

[256] Sun, X. and Li Duan, D., Optimal Condition and Branch and Bound Algorithm for Constrained Redundancy Optimization in Series System, Optimization and Engineering, 3(2002): 53-65.

[257] Gen, M. and Yun, Y., Soft computing approach for reliability optimization: State-of-the-art survey, Reliability Engineering & System Safety, 91(2006):1008-1026.

[258] Ha, C. and Kuo, W., Multi-Path heuristic for redundancy allocation: the tree heuristic, IEEE Transactions on Reliability, 55(2006): 37-43.

[259] Coelho, L. S., An efficient particle swarm approach for mixed-integer programming in reliability redundancy optimization applications, Reliability Engineering and System Safety, 94(2009): 830-837.

[260] Coelho, L. S., Self-organizing migrating strategies applied to reliability-redundancy optimization of systems, IEEE Transactions on Reliability, 58(2009): 501-510.

[261] Moore, R. E., Methods and applications of interval analysis. SIAM, Philadelphia (1979).

[262] Hansen, E. R. and Walster, G. W., Global optimization using interval analysis, Marcel Dekker Inc., New York (2004).

[263] Goldberg, David E., and John H. Holland. "Genetic algorithms and machine learning." Machine learning 3.2 (1988): 95-99.

[264] Gen, Mitsuo, and Runwei Cheng. "A survey of penalty techniques in genetic algorithms." Evolutionary Computation, 1996., Proceedings of IEEE International Conference on. IEEE, 1996.

[265] Bäck, Thomas. "Evolutionary algorithms in theory and practice." (1996).

[266] Bhunia, Asoke Kumar, Laxminarayan Sahoo, and Dilip Roy. "Reliability stochastic optimization for a series system with interval component reliability via genetic algorithm." Applied Mathematics and Computation 216.3 (2010): 929-939.

[267] Eberhart, Russ C., and James Kennedy. "A new optimizer using particle swarm theory." Proceedings of the sixth international symposium on micro machine and human science. Vol. 1. 1995.

[268] Kennedy, James. "Particle swarm optimization." Encyclopedia of machine learning. Springer US, 2011. 760-766.

[269] Clerc, Maurice. "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization." Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on. Vol. 3. IEEE, 1999.

[270] Clerc, Maurice, and James Kennedy. "The particle swarm-explosion, stability, and convergence in a multidimensional complex space." Evolutionary Computation, IEEE Transactions on 6.1 (2002): 58-73.

[271] Zhang, ShiHai, and JinPing Ou. "BP-PSO-based intelligent case retrieval method for high-rise structural form selection." Science China Technological Sciences 56.4 (2013): 940-944.

[272] Sun, Jun, Wenbo Xu, and Bin Feng. "A global search strategy of quantum-behaved particle swarm optimization." Cybernetics and Intelligent Systems, 2004 IEEE Conference on. Vol. 1. IEEE, 2004.

[273] dos Santos Coelho, Leandro. "A quantum particle swarm optimizer with chaotic mutation operator." Chaos, Solitons & Fractals 37.5 (2008): 1409-1418.

[274] Xi, Maolong, Jun Sun, and Wenbo Xu. "An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position." Applied Mathematics and Computation 205.2 (2008): 751-759.

[275] Dos Santos Coelho, Leandro, and Viviana Cocco Mariani. "Particle swarm approach based on quantum mechanics and harmonic oscillator potential well for economic load dispatch with valve-point effects." Energy Conversion and Management 49.11 (2008): 3080-3085.

[276] Liu, Jing, Qi Wu, and Daqi Zhu. "Thruster fault-tolerant for UUVs based on quantum-behaved particle swarm optimization." Opportunities and challenges for next-generation applied intelligence. Springer Berlin Heidelberg, 2009. 159-165.

[277] Storn, Rainer, and Kenneth V. Price. "Minimizing the Real Functions of the ICEC'96 Contest by Differential Evolution." International Conference on Evolutionary Computation. 1996.

[278] Vesterstrom, Jakob, and Rene Thomsen. "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems." IEEE Congress on Evolutionary Computation. Vol. 2. 2004.

[279] Dorigo, Marco, and Christian Blum. "Ant colony optimization theory: A survey." Theoretical computer science 344.2 (2005): 243-278.

[280] Dai, Qiguo, Junzhong Ji, and Chunnian Liu. "An effective initialization strategy of pheromone for ant colony optimization." Bio-Inspired Computing, 2009. BIC-TA'09. Fourth International Conference on. IEEE, 2009.

[281] Chern, Maw-Sheng. "On the computational complexity of reliability redundancy allocation in a series system." Operations research letters 11.5 (1992): 309-315.

[282] Nakagawa, Yuji, and Kyoichi Nakashima. "A heuristic method for determining optimal reliability allocation." IEEE Transactions on reliability 26.3 (1977): 156-161.

[283] Kim, Jae-Hwan, and Bong-Jin Yum. "A heuristic method for solving redundancy optimization problems in complex systems." IEEE Transactions on Reliability 42.4 (1993): 572-578.

[284] Kuo, Way, Ching-Lai Hwang, and Frank A. Tillman. "A note on heuristic methods in optimal system reliability." IEEE Transactions on Reliability 27.5 (1978): 320-324.

[285] Aggarwal, K. K., and J. S. Gupta. "Penalty function approach in heuristic algorithms for constrained." IEEE Transactions on Reliability 54.3 (2005): 549-558.

[286] Kulturel-Konak, Sadan, Alice E. Smith, and David W. Coit. "Efficiently solving the redundancy allocation problem using tabu search." IIE transactions 35.6 (2003): 515-526.

[287] Hwang, Ching-Lai, Frank A. Tillman, and Way Kuo. "Reliability optimization by generalized Lagrangian-function and reduced-gradient methods." IEEE Transactions on reliability 28.4 (1979): 316-319.

[288] Kuo, Way, et al. "Reliability optimization with the Lagrange-multiplier and branch-and-bound technique." IEEE Transactions on Reliability 36.5 (1987): 624-630.

[289] Tillman, Frank A., Ching-Lai Hwang, and Way Kuo. "Determining component reliability and redundancy for optimum system reliability." IEEE Transactions on Reliability 26.3 (1977): 162-165.

[290] Sun, Xiaoling L., and Duan Li. "Optimality condition and branch and bound algorithm for constrained redundancy optimization in series systems." Optimization and Engineering 3.1 (2002): 53-65.

[291] Sung, C.S., Cho, Y. K., Branch-and-Bound redundancy optimization for a series system with multiple-choice constraints, IEEE Transactions on Reliability 48 (1999) 108-117.

[292] Sakoe, Hiroaki, and Seibi Chiba. "Dynamic programming algorithm optimization for spoken word recognition." IEEE transactions on acoustics, speech, and signal processing 26.1 (1978): 43-49.

[293] Nakagawa, Y., Miyazaki, S., Surrogate constraints algorithm for reliability optimization problems with two constraints, IEEE Transactions on Reliability R-30 (1981) 175-180.

[294] Hikita, M., Nakagawa, Y., Narihisa, H., Reliability optimization of systems by a surrogate constraints algorithm, IEEE Transactions on Reliability 41(3) (1992) 473-480.

[295] Dengiz, B., Altiparmak, F., Smith, A.E., Local search genetic algorithm for optimal design of reliable networks, IEEE Transactions on Evolutionary Computation 1(3) (1997) 179-188.

[296] Tavakkoli-Moghaddama, R., Safar, J., Sassani, F., Reliability optimization of series-parallel systems with a choice of redundancy strategies using a genetic algorithm, Reliability Engineering and System Safety 93 (2008) 550-556.

[297] Ye, Z., Li, Z., Min, X., Some improvements on adaptive genetic algorithms for reliability-related applications, Reliability Engineering and System Safety 95 (2010) 120-126.

[298] Gupta, R.K., Bhunia, A.K., Roy, D., A GA based penalty function technique for solving constrained redundancy allocation problem of series system with interval valued reliabilities of components, Journal of Computational and Applied Mathematics 232 (2009) 275-284.

[299] Bhunia, A. K., and M. Maiti. "A two warehouse inventory model for deteriorating items with a linear trend in demand and shortages." Journal of the Operational Research Society 49.3 (1998): 287-292.

[300] Bhunia, A.K., Sahoo, L., Genetic algorithm based reliability optimization in interval environment, Innovative Computing Methods and Their Applications to Engineering Problems, N. Nedjah (Eds.).SCI 357 (2011) 13-36.

[301] Sahoo, L., Bhunia, A.K., Kapur, P.K., Genetic algorithm based multi-objective reliability optimization in interval environment, Computers and Industrial Engineering 62 (2012) 152-160.

[302] Sahoo, L., Bhunia, A.K., Roy, D., An application of genetic algorithm in solving reliability optimization problem under interval component Weibull parameters, Mexican Journal of Operations Research 1(1) (2012) 2-19.

[303] Mahato, S.K.,. Sahoo, L, Bhunia, A.K., Reliability-redundancy optimization problem with interval valued reliabilities of components via genetic algorithm, Journal of Information and Computing Science 7(4) (2012) 284-295.

[304] Sahoo, L., Bhunia, A. K., Roy, D., Reliability optimization in stochastic domain via genetic algorithm, International Journal of Quality & Reliability Management (2013).

[305] Sahoo, L., Bhunia, A.K., Roy, D., Reliability optimization with high and low level redundancies in interval environment via Genetic Algorithm, International Journal of Systems Assurance Engineering and Management DOI: 10.1007/s13198-013-0199-9 (2013).

[306] Hsieh, Y.C., Chen, T.C., Bricker, D.L., Genetic algorithms for reliability design problems, Microelectronics Reliability 38 (1998)1599-1605.

[307] Hikita, Mitsunori, et al. "Reliability optimization of systems by a surrogate-constraints algorithm." IEEE Transactions on Reliability 41.3 (1992): 473-480.

[308] Kim, H.G., Bae, C., Park, D.J., Reliability-redundancy optimization using simulated annealing algorithms, Journal of Quality in Maintenance Engineering 12(4) (2006) 354-363.

[309] Yokota,T., Gen, M., Li, Y.X., Genetic algorithm for non-linear mixed-integer programming problems and its applications, Computers and Industrial Engineering 30(4) (1996) 905-917.

[310] Coelho, L.D.S., An efficient particle swarm approach for mixed-integer programming in reliability-redundancy optimization applications, Reliability Engineering and System Safety 94 (2009) 830-837.

[311] Chen, T. C., IAs based approach for reliability redundancy allocation problems, Applied Mathematics Computation 182(2) (2006) 1556-1567.

[312] Sheikhalishahi, M., et al. "A hybrid GA-PSO approach for reliability optimization in redundancy allocation problem." International Journal of Advanced Manufacturing Technology 68 (2013).

[313] Sakawa, Masatoshi. "Multiobjective optimization by the surrogate worth trade-off method." *IEEE Transactions on Reliability* 27.5 (1978): 311-314.

[314] Inagaki, Toshiyuki, Koichi Inoue, and Hajime Akashi. "Interactive optimization of system reliability under multiple objectives." *IEEE Transactions on Reliability* 27.4 (1978): 264-267.

[315] Park, Morag, et al. "Sequence of MET protooncogene cDNA has features characteristic of the tyrosine kinase family of growth-factor receptors." *Proceedings of the National Academy of Sciences* 84.18 (1987): 6379-6383.

[316] Dhingra, Anoop K. "Optimal apportionment of reliability and redundancy in series systems under multiple objectives." *IEEE Transactions on reliability* 41.4 (1992): 576-582.

[317] Rao, S. S., and A. K. Dhingra. "Reliability and redundancy apportionment using crisp and fuzzy multiobjective optimization approaches." *Reliability Engineering & System Safety* 37.3 (1992): 253-261.

[318] Rao, S. S., and A. K. Dhingra. "Reliability and redundancy apportionment using crisp and fuzzy multiobjective optimization approaches." *Reliability Engineering & System Safety* 37.3 (1992): 253-261.

[319] Huang, Hong-Zhong, Zhigang Tian, and Ming J. Zuo. "Intelligent interactive multiobjective optimization method and its application to reliability optimization." *IIE Transactions* 37.11 (2005): 983-993.

[320] Coit, David W., and Abdullah Konak. "Multiple weighted objectives heuristic for the redundancy allocation problem." *IEEE transactions on reliability* 55.3 (2006): 551-558.

[321] Taboada, Heidi A., and David W. Coit. "Data clustering of solutions for multiple objective system reliability optimization problems." *Quality Technology & Quantitative Management* 4.2 (2007): 191-210.

[322] Taboada, Heidi A., Jose F. Espiritu, and David W. Coit. "MOMS-GA: A multi-objective multi-state genetic algorithm for system reliability optimization design problems." *IEEE Transactions on Reliability* 57.1 (2008): 182-191.

[323] Taboada, H. A., J. F. Espiritu, and D. W. Coit. "Design allocation of multistate series-parallel systems for power systems planning: A multiple objective evolutionary approach." *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 222.3 (2008): 381-391.

[324] Li, Zhaojun, Haitao Liao, and David W. Coit. "A two-stage approach for multi-objective decision making with applications to system reliability optimization." *Reliability Engineering & System Safety* 94.10 (2009): 1585-1592.

[325] Ramirez-Marquez, Jose Emmanuel, and Claudio M. Rocco. "Evolutionary optimization technique for multi-state two-terminal reliability allocation in multi-objective problems." *IIE Transactions* 42.8 (2010): 539-552.

[326] Rocco, C. M., et al. "Implementation of multi-objective optimization for vulnerability analysis of complex networks." *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 224.2 (2010): 87-95.

[327] Dewri, Rinku, et al. "Optimal security hardening using multi-objective optimization on attack tree models of networks." *Proceedings of the 14th ACM conference on Computer and communications security*. ACM, 2007.

[328] Prasad, V. Rajendra, and Way Kuo. "Reliability optimization of coherent systems." *IEEE Transactions on Reliability* 49.3 (2000): 323-330.

[329] Levitin, Gregory, and Anatoly Lisnianski. "A new approach to solving problems of multi-state system reliability optimization." Quality and reliability engineering international 17.2 (2001): 93-104.

[330] Kuo, Way, et al. *Optimal reliability design: fundamentals and applications*. Cambridge university press, 2001.

[331] Miettinen, Kaisa. "A Posteriori Methods." *Nonlinear multiobjective optimization*. Springer, Boston, MA, 1998. 77-113.

[332] Charnes, Abraham, and William Wager Cooper. "Goal programming and multiple objective optimizations: Part 1." *European Journal of Operational Research* 1.1 (1977): 39-54.

[333] Mainwaring, Alan, et al. "Wireless sensor networks for habitat monitoring." *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. Acm, 2002.

[334] Arampatzis, Th, John Lygeros, and Stamatis Manesis. "A survey of applications of wireless sensors and wireless sensor networks." *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*. IEEE, 2005.

[335] Cartina, G., et al. "Improved fuzzy load models by clustering techniques in optimal planning of distribution networks." PowerTech, 2009 IEEE Bucharest. IEEE, 2009.

[336] Moscal, Dinu. "Dumitru Loșonți, Certitudini și ipoteze etimologice, Editura Academiei Române, București, 2007, 187 p." Diacronia 2015.3 (2015): A44-A44.

[337] Zhao, Xinchao, Xiao-Shan Gao, and Ze-Chun Hu. "Evolutionary programming based on non-uniform mutation." Applied Mathematics and Computation 192.1 (2007): 1-11.

[338] Pastorino, Matteo. "Stochastic optimization methods applied to microwave imaging: A review." IEEE Transactions on Antennas and Propagation 55.3 (2007): 538-548.

[339] Grosan, Crina, and Ajith Abraham. "Hybrid evolutionary algorithms: methodologies, architectures, and reviews." Hybrid evolutionary algorithms. Springer Berlin Heidelberg, 2007. 1-17. doi:10.1007/978-3-540-73297-6_1

[340] Tripathi, Praveen Kumar, Sanghamitra Bandyopadhyay, and Sankar Kumar Pal. "Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients." Information Sciences 177.22 (2007): 5033-5049.

[341] Lee, Zne-Jung, et al. "Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment." Applied Soft Computing 8.1 (2008): 55-78.

[342] Levitin, Gregory, Shmuel Mazal-Tov, and David Elmakis. "Genetic algorithm for optimal sectionalizing in radial distribution systems with alternative supply." Electric Power Systems Research 35.3 (1995): 149-155.

[343] Quiroga, O. A., J. Meléndez, and S. Herraiz. "Fault causes analysis in feeders of power distribution networks." International Conference in Renewables Energies and Quality Power, ICREP. Vol. 11. 2011. doi:10.24084/repqj09.619.

[344] Neagu, B. O. G. D. A. N., Gh Georgescu, and M. Gusa. "Load Curves Characteristics of Consumers Supplied From Electricity Repartition and Distribution Public Systems." The Bulletin of the Polytechnic Institute of Iasi, Tomul LVII (LXI), Fasc 1 (2011): 141-157.

Signature of the Candidate
(Avishek Banerjee)