APPLICATION OF GENETIC ALGORITHM BASED OPTIMIZATION METHODOLOGIES TO SOME OF THE SYSTEMS OF INDIAN NUCLEAR REACTORS

By

M. L. JAYALAL

(ENGG02200804016)

Indira Gandhi Center for Atomic Research
Kalpakkam-603102, Tamil Nadu, India

A thesis submitted to the
Board of Studies in Engineering Sciences
In partial fulfillment of requirements
for the Degree of

DOCTOR OF PHILOSOPHY

of

HOMI BHABHA NATIONAL INSTITUTE

August 2015

Homi Bhabha National Institute

Recommendations of the Viva Voce Committee

As members of the Viva Voce Committee, we certify that we have read the dissertation prepared by M.L. Jayalal entitled "**Application of Genetic Algorithm Based Optimization Methodologies to Some of the Systems of Indian Nuclear Reactors**" and recommend that it may be accepted as fulfilling the thesis requirement for the award of Degree of Doctor of Philosophy.

Chairman	Dr. B. Purna Chandra Rao	Date: 8/4/16
Convener	Dr. M. Sai Baba	Date: 8-4-16
External Examiner	Prof. G. Sivakumar	Date: 8/4/16
Member 1	Dr. B.V.R. Tata	Date: 8/4/16
Member 2	Dr. B.K. Panigrahi	Date: 8/4/16
Technology Advisor	Dr. S. A. V. Satya Murty	Date: 8/4/16

Final approval and acceptance of this thesis is contingent upon the candidate's submission of the final copies of the thesis to HBNI.

I hereby certify that I have read this thesis prepared under my direction and recommend that it may be accepted as fulfilling the thesis requirement.

Date: 08 April, 2016
Place: Kalpakkam

Guide: Dr. M. Sai Baba

STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at Homi Bhabha National Institute (HBNI) and is deposited in the Library to be made available to borrowers under the rules of the HBNI.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the Complement Authority of HBNI when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

M.L. Jayalal

DECLARATION

I, hereby declare that the investigation presented in the thesis has been carried out by me. The work is original and has not been submitted earlier as a whole or in part for a degree/ diploma at this or any other Institution / University.

M.L. Jayalal

LIST OF PUBLICATIONS

Journal

1. Steam Condenser Optimization using Real Parameter Genetic Algorithm for Prototype Fast Breeder Reactor

   M.L. Jayalal, L. Satish Kumar, R. Jehadeesan, S. Rajeswari, S.A.V. Satya Murty, V. Balasubramaniyan, S.C. Chetal

   Nuclear Engineering and Design 241(2011) 4136-4142

2. Computational Intelligent Systems for Prototype Fast Breeder Reactor

   M.L. Jayalal, T. Jayanthi, S.A.V. Satya Murty, P. Swaminathan

   Energy Procedia 7(2011) 589-596

3. A Survey of Genetic Algorithm Applications in Nuclear Fuel Management

   M.L. Jayalal, S.A.V. Satya Murty, M. Sai Baba

   Journal of Nuclear Engineering and Technology 4(2014) 45-62

4. Application of Genetic Algorithm Methodologies in Fuel Bundle Burnup Optimization of Pressurized Heavy Water Reactor

   M.L. Jayalal, Suja Ramachandran, S. Rathakrishnan, S.A.V. Satya Murty, M. Sai Baba

   Nuclear Engineering and Design, 281 (2015) 58–71

Conferences

1. Application of Genetic Algorithms in Data Mining and Knowledge Acquisition

   M.L.Jayalal, S.Rajeswari, S.A.V.Satya Murty

   8th biennial national conference on Recent Advances in Information Technology (READIT-2011) held at IGCAR, Kalpakkam (2011)

2. Knowledge Discovery Using Genetic Algorithm: A Study

   M.L.Jayalal, M. Sai Baba, S.Rajeswari, K.K. Kuriakose, S.A.V.Satya Murty

   9th biennial national conference on Recent Advances in Information Technology (READIT-2013) held at IGCAR, Kalpakkam (2013)

DEDICATIONS

I dedicate this thesis

to my

Father, Mother, Wife and Son

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to my guide, Dr. M. Sai Baba, for being a great teacher and source of inspiration to me. It was because of his continuous guidance and all round support that led to the successful completion of this work.

I am extremely thankful to my technology advisor, Dr. S.A.V. Satya Murty for his encouraging words and constant support at each stage of the research.

It is my great privilege to acknowledge members of the doctoral committee, Dr. B. Purna Chandra Rao (Chairman), Dr. B.V.R. Tata and Dr. B.K. Panigrahi, for their critical review and valuable comments. I thank Dr. T. Jayakumar (former Dean-Academic, HBNI) and Dr. G. Sasikala (Dean-Academic, HBNI), for the support received from them. I am indebted to many of my senior colleagues, particularly, Mrs. S. Rajeswari, Dr. K. Devan and Mr. S. Rathakrishnan and Mr. M.C. Valsa Kumar, for all the helps and advices. I would like to thank my friends and colleagues, Mr. L. Satish, Dr. Riyas, Mr. V. Vinod, Mr. J.K. Sahu and Mrs. Suja Ramachandran, for the useful technical discussions with them, during the course of this work. I would like to extend my gratitude to Prof. S. Thiruvengadam, for his valuable suggestions in writing the thesis.

I am very grateful to all the members of computer division, who have been directly or indirectly related to this work, for their help and support throughout these years. I would like to specially thank Mr. R. Nanda Kumar, Mr. V. Balu, Mr.N. Satish Kumar, Mr. A. Boopalan, Mr. Ibrahim Khan, and Mr. K. Rajesh, for their helps. I am heartily thankful to Dr. D. Thirugnanamuthy, Dr. Vidhya Sundararajan, Dr. D. Sunil Kumar and Mr. Joel Jose, for the helps they lent to me during my research work.

Finally, I express my gratitude to all persons who have not been listed here, but helped me in different ways in completing this work.

CONTENTS

Page No.

CHAPTER 2

A STUDY ON GENETIC ALGORITHMS 25-61

CHAPTER 3

APPLICATION OF GENETIC ALGORITHM IN STEAM CONDENSER OPTIMIZATION OF PROTOTYPE FAST BREEDER REACTOR 63-89

CHAPTER 4

A STUDY ON GENETIC ALGORITHM METHODOLOGIES IN FUEL BUNDLE BURNUP OPTIMIZATION OF PRESSURIZED HEAVY WATER REACTOR 91-122

SYNOPSIS

1. Preamble

Computational Intelligence (CI) is one of the major domains of artificial intelligence that deals with adaptive mechanisms to enable or facilitate intelligent behaviour in complex and changing environment. The three main pillars of Computational Intelligence are Artificial Neural Networks, Fuzzy Logic Systems and Genetic Algorithms. Computational Intelligence techniques have been successfully employed in a wide range of applications which include the domains of medicine, bioinformatics, electronics, communications and business. Even though there has been certain progress during the last two decades, the potential for the application of Computational Intelligence in the nuclear domain has not been explored to significant extent. The stringent nuclear safety regulations pertaining to reactor environment, poses additional challenge in the application of Computational Intelligence in various nuclear systems.

Genetic Algorithm is one of the main Computational Intelligence paradigms where the elements of Darwin’s theory of natural evolution are modelled algorithmically. Genetic Algorithms have been shown to be more efficient than classical optimization algorithms (like gradient based methods and simplex methods) for discontinuous, non-differentiable and multimodal problems. The advantages of Genetic Algorithm offer considerable scope for its applications in nuclear reactor related optimization problems. In fact, one of the early practical applications of Genetic

Algorithm was in nuclear reactor environment, specifically in nuclear fuel management of Pressurized Water Reactors.

The literature survey carried out as the part of this work indicates that there are potential areas in nuclear engineering filed which are little explored for the application of an intelligent optimization technique like Genetic Algorithm. For example, even though there has been progress in the application of Genetic Algorithms in nuclear fuel management of Pressurized Water Reactors and Boiling Water Reactors, similar applications in Pressurized Heavy Water Reactors (PHWRs) and Fast Breeder Reactors (FBRs) were found to be limited. These two types of reactors are the important building blocks of the three-stage nuclear power programme of India. Therefore, the focus of the work carried out as part of the thesis is on the application of Genetic Algorithm based optimization methodologies in PHWRs and FBRs.

2. Motivation for the Research

The main motivation behind the presented work is the application and comparison of Genetic Algorithm based optimization methodologies in the selected reactor systems, to arrive at the suitability of a methodology for a particular type of application. There is a scope for exploring the possibilities of a modular approach in developing such applications, which helps in extending the Genetic Algorithm based optimization methodologies to other reactor systems.

One of the potential areas of application of Genetic Algorithm is in the optimization problems of nuclear fuel management. The nuclear fuel management optimizations represent a range of optimal decision making problems, from the amount and physical properties of the fuel inventory to the loading patterns of fuel elements. The general objective is to maximize reactor performance or improve the flexibility of reactor operation, subjected to operational and safety constraints. The advanced tools, based on intelligent optimization techniques like Genetic Algorithms, are of great potential in this field. The present status of Genetic Algorithm based methodologies in nuclear fuel management is surveyed in the work carried out as part of the thesis and arrived at two relevant methodologies applicable for the study. The methodologies selected for the study are, Penalty Functions based Genetic Algorithm (referred to as Penalty-function GA) and Multi Objective Genetic Algorithm (referred to as Multi-objective GA). In the case of Penalty-function GA, the multi-objective optimization problem is converted in to single objective by adding penalty functions and constraints. Multi-objective GA handles the multiple objectives altogether, by incorporating the concepts of Pareto-optimality and dominance. The evaluation of these methodologies for reactor applications is a novel initiative highlighted in this work. The study also explores in identifying the appropriate choice of Genetic Algorithm methodology for a particular reactor application.

3. Description of the Work

A modular approach has been employed in the development and implementation of the optimization methodologies. The development procedure employed in the study is divided in to three modules: (i) Genetic Algorithm module (ii) interface module (iii) reactor process simulation module. The development of Genetic Algorithm module involves the mathematical formulation of the reactor based optimization problem and the development of computer codes (using 'C' programming language) for implementing different Genetic Algorithm methodologies to obtain the solutions. The interface module acts as a communication channel between the Genetic Algorithm module and the reactor process simulation module. The interface module developed (using 'C' and 'R' programming languages) for a particular reactor application is unique and capable of automated iterative execution. This approach has provided the flexibility in modifying the modules for extending the applications to other reactor systems.

The optimization studies carried out as part of the thesis include, design optimization problem of steam condenser of 500 MWe Prototype Fast Breeder Reactor (PFBR), fuel bundle optimization of a 220 MWe natural uranium fuelled Pressurized Heavy Water Reactor (PHWR), optimal core configuration for Prototype Fast Breeder Reactor (PFBR) having two different fuel enrichment zones and optimal core configuration for 1000 MWe Fast Breeder Reactor having three different fuel enrichment zones. The first study is an engineering design optimization application and the other three are of nuclear fuel management. The standard real-parameter Genetic Algorithm model is chosen for the steam condenser optimization study. The Penalty-function GA and Multi-objective GA are applied and evaluated in the nuclear fuel management studies. The studies carried out as part of the thesis are briefly described in the following subsections.

3.1. Steam Condenser Optimization

The subsystem considered for the optimization study is the steam condenser (or circulating water system) of PFBR. The study is an engineering design optimization problem. The purpose of the study is to apply standard real-parameter Genetic Algorithm for single objective optimization problem of reactor systems. The determination of feasible values for the design parameters of the condenser, that influence the system performance, is the aim of the study. Then, Genetic Algorithm based performance-cost analysis is carried out to find the optimum circulating water system design, based on the maximum capitalized profit. The design parameters considered are: flow rate of condenser, outer diameter of condenser tube, length of condenser tube and velocity of water inside the tube. The optimization problem has the objective of maximization of the capitalized profit and has to satisfy two constraints. The first constraint is related to the temperature range of the circulating water system and the second one is related terminal temperature difference of the condenser. The results obtained are validated by comparing with the results available based on the conventional optimization study.

3.2. Fuel Bundle Burnup Optimization of Pressurized Heavy Water Reactor (PHWR)

The fuel bundle optimization of PHWR involves, finding the arrangement of fresh and partially burned fuel bundles within the reactor core that optimizes the performance of the reactor, while ensuring that operational and safety features are satisfied. The aim of the study is to calculate the optimum discharge burnups of the fuel bundles which give maximum reactor power without violating various constraints, such as maximum bundle power and maximum channel power. In the presented work, the number of burnup zones of the reactor core is fixed as two; an inner zone of high burnup and an outer zone of low burnup. The objectives selected for the optimization are: maximization of the average discharge burnup, maximization of the effective multiplication factor, minimization of the maximum bundle power and minimization of the maximum channel power. The discharge burnups arrived by the Genetic Algorithm based optimization methodologies can be utilized, in fixing the most suitable reference discharge burnups for the two burnup zones of the reactor core. Two optimization methodologies, i.e. the Penalty-function GA and the Multi-objective GA, have been applied and compared.

3.3. Core Configuration Optimization Studies on Fast Breeder Reactors

The studies presented on optimal core configuration of Fast Breeder Reactor cores explore the machine learning and computational intelligence abilities associated with Genetic Algorithms in finding the optimal number of subassemblies in the reactor core to achieve the best performance of the reactor. Finding out optimal core configuration of Fast Breeder Reactor, is the result of detailed neutronics scoping studies, taking into consideration of several factors like, size of the core, enrichment of the fuel, linear heat rating of the fuel pins, excess reactivity of the core, control rod design, and the inventory of the fuel. Therefore, optimization of the core configuration design is a complex task in terms of computational effort and time. Two different Fast Breeder Reactor cores, (i) 500 MWe core having two different fuel enrichment zones (ii) 1000 MWe core having three different fuel enrichment zones, are considered in the studies. The two reactor cores are different in their size, generated power, number of subassemblies present in the core, types of fuel, fuel enrichments and number of enrichment zones present.

The objectives selected for the optimization (in both cases) are related to linear heat rate of the enrichment zones, excess reactivity of the core, breeding ratio achieved for the configuration and the required fuel inventory. The linear heat rating is the power generated per unit length of the fuel pin. The objective is to limit its value such that the temperature in the fuel pin does not exceed the melting point of the fuel. The excess reactivity of the core indicates the effective neutron multiplication factor to be provided in the core in order to override all the reactivity losses during an operational cycle. Higher breeding ratio is desirable to generate enough fissile material for self-sufficient closed fuel cycle of Fast Breeder Reactor programme. It is also desirable to attain core configuration with possible minimum fuel inventory, which has an impact on the fuel economy.

Two optimization methodologies, i.e. the Penalty-function GA and the Multi-objective GA have been applied and compared. The results obtained from the study are verified with the reference core configurations. A modular approach has been employed in the implementation of the optimization methodologies.

4. Summary

The study carried out in the presented work was motivated by the need to develop and study, appropriate optimization methodologies based on Genetic Algorithms for the applications in diverse reactor environments. As part of the study, appropriate Genetic Algorithm based methodologies have been applied and compared in a set of diverse and less explored reactor environments of Pressurized Heavy Water Reactor and Fast Breeder Reactor. The major findings from the study are:

- The suitability of the selected Genetic Algorithm methodologies namely, Penalty-function GA and Multi-objective GA, are verified for different types of optimization problems of nuclear fuel management.
- The Multi-objective GA is found to be better than the Penalty-function GA in two important aspects: one, in terms of the diversity of generated solutions and two, in the speed of convergence of the algorithm.
- The suitability of standard real-parameter Genetic Algorithm is verified for the engineering design optimization of reactor subsystem with single objective and limited number of constraints.
- A modular approach has been employed, in applying and comparing different methodologies based on Genetic Algorithm, in the optimization problems of nuclear fuel management.

5. Future Scope of the Work

The results obtained from the studies carried out as part of the thesis clearly establishes the suitability of the selected Genetic Algorithm methodologies, in the optimization studies related to different nuclear reactor systems. The modular approach followed in the present studies, allow extending the application of the Genetic Algorithm methodologies to many other optimization studies of nuclear fuel management. In the direction of extending the applications, the following areas of study can be considered: (i) fuel bundle optimization of Pressurized Heavy Water Reactor core with different geometries of inner and outer burnup zones (ii) finding out the optimal control rod positions in the Fast Breeder Reactor core (iii) fuel enrichment optimization in different zones of the Fast Breeder Reactor core. The major obstacle in extending the Genetic Algorithm application is the computational time requirement associated with the running of process simulation codes of reactors. A solution for this is the development and application of "parallel" Genetic Algorithms (genetic algorithms developed by employing parallel programming concepts) which are capable of running in parallel computers.

6. Organization of the Thesis

The thesis is divided in to six chapters. In the Chapter 1, a brief introduction to the research work is given. The description and findings from the literature survey carried out in the field of Computational Intelligence methods applicable in different nuclear systems, including the applications of Genetic Algorithm in nuclear fuel management, are given. Chapter 2 introduces the Genetic Algorithm operators and parameters, used in the studies carried out as part of the thesis. The two methodologies of Genetic Algorithm which are employed in the optimization studies of nuclear fuel management, i.e. Penalty-function GA and Multi-objective GA, are discussed in detail.

Chapter 3 addresses the engineering optimization study conducted on steam condenser of PFBR. Chapter 4 explains the work carried out for the fuel bundle optimization of PHWR. Chapter 5 covers the two different optimization studies carried out on core configuration of Fast Breeder Reactors. Chapter 6 summarizes the work carried out in the thesis, the conclusions are drawn from the studies carried out and possible areas of future work.

LIST OF FIGURES

LIST OF TABLES

TABLE OF ABBREVIATIONS

Symbol	Abbreviations
CI	Computational Intelligence
AI	Artificial Intelligence
GA	Genetic Algorithm
ANN	Artificial Neural Network
FLS	Fuzzy Logic Systems
IGCAR	Indira Gandhi Centre for Atomic Research
FBR	Fast Breeder Reactor
PWR	Pressurized Water Reactor
BWR	Boiling Water Reactor
PHWR	Pressurized Heavy Water Reactor
GCR	Gas Cooled Reactor
PFBR	Prototype Fast Breeder Reactor
CR	Crossover Rate
MR	Mutation Rate
EL	Elite Members
PPF	Power Peaking Factor
NSGA-II	Non-dominated Sorting Genetic Algorithm-II
CWS	Circulating Water System
TTD	Terminal Temperature Difference
MBP	Maximum Bundle Power
MCP	Maximum Channel Power
CSR	Control and Safety Rods
DSR	Diverse Safety Rods

CHAPTER 1

INTRODUCTION

This chapter gives an introduction to the major disciplines connected to the area of Computational Intelligence, nuclear reactors and the applications of Computational Intelligence in nuclear reactors. The literature survey carried out in the area of application of Genetic Algorithms in nuclear fuel management and the findings from the survey are presented.

The increasing demand for intelligent search, optimization and machine learning in several applications opened up new avenues for the use of Computational Intelligence methods. These methods have been successfully employed in a wide range of applications which include the domains of medicine, bioinformatics, electronics, communications and business. Even though there has been certain progress during the last two decades, the potential for the application of Computational Intelligence in the nuclear reactor domain is yet to be thoroughly explored. The stringent nuclear safety regulations pertaining to reactor environment poses additional challenge in the application of Computational Intelligence to various nuclear subsystems.

Genetic Algorithm is one of the major optimization methods coming under the umbrella of Computational Intelligence. Genetic Algorithms have been shown to be more efficient than classical optimization algorithms (like gradient based methods and simplex methods) for discontinuous, non-differentiable and multimodal problems. The advantages of Genetic Algorithm offer scope for its application to optimization problems relating to nuclear reactors. The work presented in the thesis explores the application of Genetic Algorithm based optimization methodologies to a set of diverse and less explored subsystems of nuclear reactors. An introduction to Computational Intelligence covering its features, major constituents and applications are outlined in the next section.

1.1. COMPUTATIONAL INTELLIGENCE

Computational Intelligence (CI) being a relatively new area, its identity and definition are still subjects of debate. Computational Intelligence and Artificial Intelligence (AI) are closely interlinked domains but have certain differences among them. The traditional problem solving methods in Artificial Intelligence (AI) are more concerned with symbolic representation of problem states and construction of a set of rules to describe transitions in the problem states [1, 2]. Computational Intelligence differs from the traditional Artificial Intelligence in few aspects, such as:

(i) its use of sub-symbolic knowledge i.e., working with numerical (low-level) data

(ii) having a pattern recognition component

(iii) showing tolerance for errors and noises

(iv) exhibit computational adaptability

The traditional Artificial Intelligence is not very competent to handle a vide category of machine learning problems. In order to address several real life problems, where some sort of “intelligence” is required in arriving at feasible solutions, a set of Computational Intelligence models and tools have been developed. These collection of tools and methods of Computational Intelligence are coming under the fabric of "soft computing" [3]. They are often designed to mimic one or more aspects of the intelligence of biological systems.

The major members coming under the broad discipline of Computational Intelligence are outlined in Figure 1. The figure represents only a part of the family tree of Computational Intelligence, since more and more computational methods are being included as members of this domain. The Computational Intelligence can be subdivided into child nodes such as Granular Computing, Neuro Computing, Evolutionary Computing and Artificial Life [1]. The Granular Computing deals with the processing of complex information entities called information granules, which arise in the process of data abstraction and derivation of knowledge from information or data. The major constituents of Granular Computing are fuzzy sets, rough sets and probabilistic reasoning.

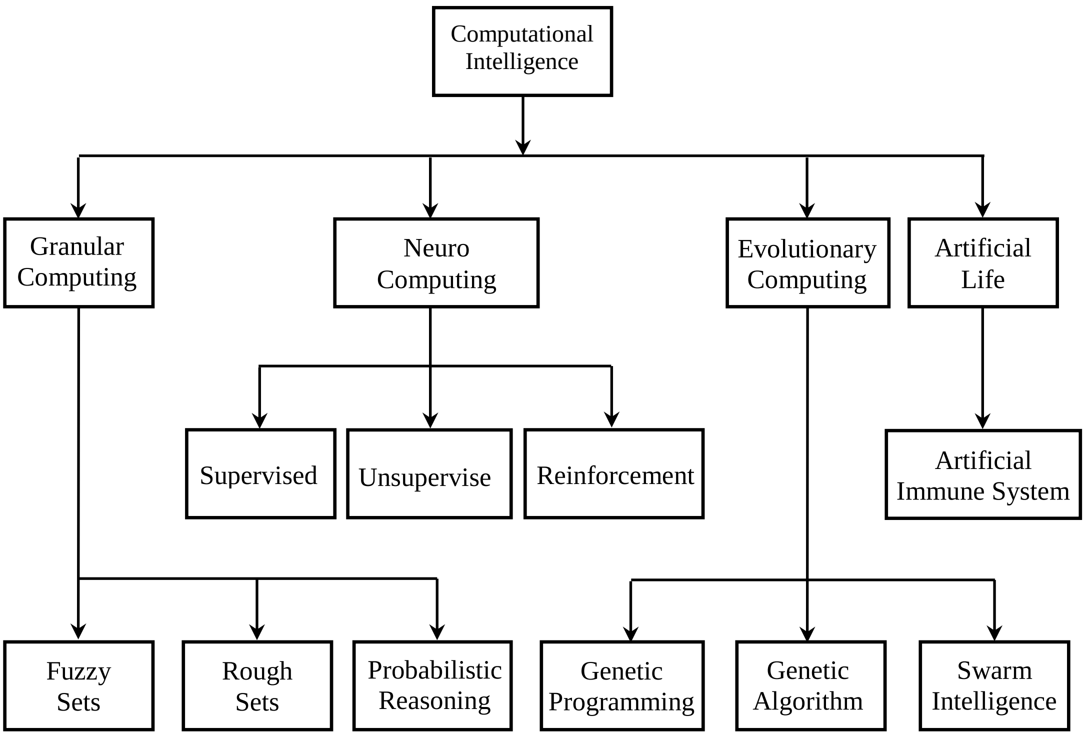


Figure 1.1: Computational Intelligence family tree showing the major members of the domain

The next child node i.e., Neuro Computing includes all the machine learning methods related to the neural networks. Evolutionary Computing is the collective name for a range of problem-solving methods based on principles of biological evolution. The Genetic Algorithm - the tool used in the studies presented in the thesis - is coming under the category of Evolutionary Computing. Artificial Life is another emerging discipline that is based on the assumption that physical and chemical laws are good enough to explain intelligence of the living organisms. As the names indicate, the Evolutionary Computing and the Artificial Life are closely related fields where the main difference lies in defining the fitness of the agents (agent represents the solution candidates which behave in accordance with the environment and the goal).

Even though the comprehensive list of members coming under the family of Computational Intelligence is large, there are three primary members, which are:

(i) Artificial Neural Networks (ANN; also referred to as neural networks in the rest of the thesis)

(ii) Fuzzy Logic Systems (FLS; also referred to as fuzzy logic in the rest of the thesis)

(iii) Evolutionary algorithms like Genetic Algorithms (Genetic Algorithm is referred to as GA in the rest of the thesis)

Each of the above-mentioned paradigms of Computational Intelligence has its origin in biological systems. Neural networks model biological neural systems, Evolutionary Algorithms model natural evolution, and fuzzy logic evolved from studies of organism’s interaction with their environment [2].

Computational Intelligence methods have been successfully employed in a wide range of applications which include engineering design [4], scheduling of factory processes [5], robots for hazardous environments [6], autonomous vehicles [7], intelligent information retrieval [8] and natural language translation [3, 9]. In fact, the domains of applications vary from agriculture and farming to control of modern power systems [10]. There are several examples in which more than one Computational Intelligence tools are applied in tandem, in order to solve certain complex tasks. One such example is the application of neural network and genetic algorithm together for extracting decision trees from a trained network [11]. Several new classes of algorithms are also emerging under the domain of Computational Intelligence. Majority of them are applied in machine learning methods like classification, regression and ranking. One such example is the bipartite ranking algorithm that is applied in identifying genes related to diseases like cancer [12, 13].

As the work carried out in the thesis is related to application of Computational Intelligence to the domain of nuclear reactors, a brief introduction to the nuclear reactors and their role in the Indian energy scenario are presented next.

1.2. NUCLEAR REACTORS

Nuclear reactors have been a part of the world's electrical energy generation system since 1950s with almost no greenhouse gas emissions. According to the World Nuclear Association (WNA), there are over four hundred commercial nuclear power reactors, being operated in 31 countries across the world with about 11% share of world’s electricity production [14]. Nuclear power is one of the environmentally benign ways of producing electricity on a large scale to meet the increasing energy demands of the world. Having gained decades of operational experience and benefiting from the ongoing development, use of nuclear reactors is a clearly viable option, for the fulfillment of future power demands of a developing country like India.

Nuclear reactors can be classified in various ways, but one general classification is based on the speed of neutrons causing most of the fission in the given reactor [15]. Accordingly, nuclear reactors can be classified in to two: thermal reactors and fast reactors. The core of thermal reactors contains a considerable portion of moderator materials that decrease the energy of neutrons to the thermal region. The core of fast reactors contains coolants with low neutron moderation property and the majority of fissions are produced by fast neutrons. Fast reactors need higher enrichment, i.e. fraction of fissile material in the fuel, than thermal reactors. The other features of fast reactors include lower pressure in the reactor vessel, higher inlet/outlet temperatures of coolant and the higher thermal efficiency [16]. The **Fast Breeder Reactors** (referred to as FBR) are a special category of fast reactors, in which more fissile material is produced by neutron capture than the fissile material consumed. The FBRs are of major interest for the Indian nuclear programme. They are referred to as Liquid Metal Fast Breeder Reactor (LMFBR), with the liquid sodium as coolant.

The thermal reactors can be further classified as:

- **Pressurized Water Reactors (PWR)**: They are cooled and moderated by high-pressure liquid water. These reactors use a pressure vessel to contain the nuclear fuel, control rods, moderator and coolant.
- **Boiling Water Reactors (BWR)**: They are cooled and moderated by water, but at a lower pressure, which allows the water to boil inside the pressure vessel producing the steam that runs the turbines.
- **Pressurized Heavy Water Reactors (PHWR)**: They are cooled and moderated by heavy water and fuelled by natural uranium. The reactor design originated in Canada and they are often referred to as CANDU (CANada Deuterium Uranium) reactors.

- **Gas Cooled Reactors (GCR)** and **Advanced Gas Cooled Reactors (AGR)**: These are generally moderated by graphite and cooled by carbon dioxide. They can have a high thermal efficiency compared with PWRs due to higher operating temperatures.

Among the different types of nuclear reactors mentioned, the Fast Breeder Reactors and Pressurized Heavy Water Reactors got special significance in the nuclear power programme of India. Homi Bhabha, father of Indian nuclear programme, envisaged a three-stage nuclear power programme for India which is discussed briefly in the next section.

1.3. THE THREE STAGE NUCLEAR POWER PROGRAMME OF INDIA

The emphasis of the three-stage nuclear power programme is on self-reliance and judicious utilization of country's limited uranium resources and effective utilization of vast thorium resources [17, 18]. The schematic of the Indian three-stage nuclear power programme is shown in Figure 1.2. Natural uranium fuelled Pressurized Heavy Water Reactors (PHWRs) are being operated in the first stage. The majority of the operating nuclear power plants in India are based on PHWR, which can efficiently produce the fissile material required for the country's second stage of nuclear programme. In the second stage, plutonium and the depleted uranium from PHWR would be utilized in the Fast Breeder Reactors (shown as "Pu Fuelled Breeders" in the figure). During this stage, thorium will be used in the blanket for breeding U^{233}.The third stage is based on the thermal breeder reactors (shown as "U^{233} Breeders" in the figure), where the thorium and the U^{233} (bred from the FBRs) will be used in the core as fuel. This in turn will use large-scale thorium-uranium fuel cycle in the third stage.

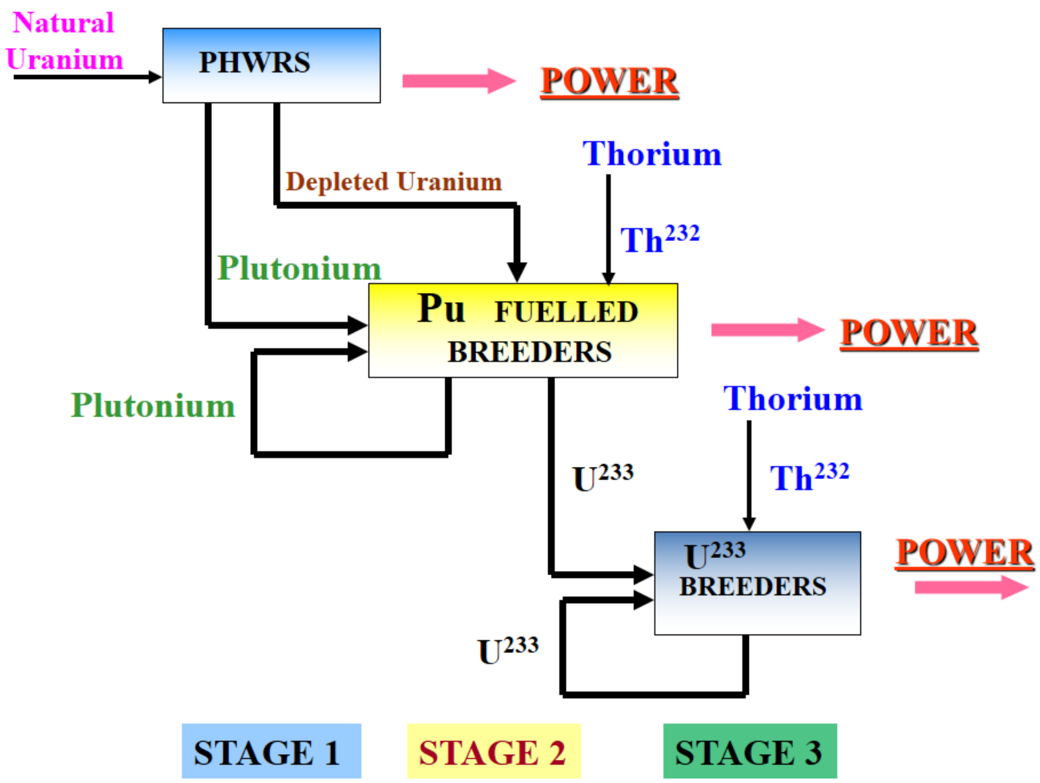


Figure 1.2: Schematic of the three-stage nuclear power programme of India

Since the development of nuclear power envisages the design and development of newer reactors, the scope for application of Computational Intelligence methods in the domain is found to be promising. A literature survey was carried out in the area of applications of Computational Intelligence in the domain of nuclear reactors and the major findings are presented next.

1.4. APPLICATION OF COMPUTATIONAL INTELLIGENCE IN THE DOMAIN OF NUCLEAR REACTORS

Applications of Computational Intelligence (considering its primary members i.e., Neural networks, Fuzzy logic systems and genetic algorithms) in the domain of nuclear reactors are categorized in general as:

(i) Neural networks which are suitable for parameter estimation or prediction

(ii) Fuzzy logic systems which are used in the applications related to the plant control

(iii) Genetic algorithms for use in the applications related to optimization

There are many hybrid systems where methods from these different paradigms are combined to solve complex problems. In the following discussion, some of the applications of hybrid systems, in the domain of nuclear reactors, are also described.

1.4.1. Application of Artificial Neural Network (ANN) in Reactor Environment

ANN is an algorithmic model of biological neural systems, which is capable of machine learning as well as pattern recognition. Neural networks can automatically learn to recognize patterns in data from real systems, from physical models, from computer programs or from other sources. They can learn new associations, new functional dependencies and new patterns [19]. The applications of neural networks cover a wide range, including diagnosis of diseases, speech recognition, data mining, pattern recognition, image processing, manufacturing, marketing and finance [20].

One of the major applications of neural networks in the domain of nuclear reactors is in **transient identification**. The early identification of unexpected departures from steady state behavior i.e. transients, is an essential step for the safe operation, control and accident management in nuclear reactors. The neural network based system for transient identification is basically a pattern recognition system utilizing neural network modules to detect the patterns corresponding to the transients [21]. One of the early applications of neural networks in transient identification of nuclear reactor was by Bartlett and Uhrig [22]. They applied the concept in identifying a set of transients related to loss of coolant, ejection of control rods and tube leak in steam generators. A novel model of two level classifier systems based on neural network was proposed by Mo and colleagues [23]. According to their model, the input signals from the nuclear reactor are fed to the first level of the system which recognizes the type of the transient. The results of the first level are sent to the second level where each transient type is handled by a separate ANN module. In the second level, the severity and the location of the transient are analyzed and detailed information about the transients is provided to the operators.

Another application of neural networks is in the **power prediction and control** of the reactors. The reactor power control is usually done using control rods, made up of neutron absorbing materials. In such situation, it is useful for the reactor operator to know in advance, how the reactor power is varying in accordance with the control rod movement. The neural network model can be trained with the reactor power at different positions of the control rod. A well-trained ANN model can predict the reactor power instantly at various operational stages, with control rods at different positions. Models based on the neural networks for online power prediction and control of reactor was investigated by Roh et al. [24] and Arab-Alibeik et al. [25].

Another well-investigated application of neural network by many researchers is in the **noise analysis** during the operation of nuclear reactor. Reactor noise analysis can be used mainly in fault detection and diagnosis of various components of nuclear reactors [26]. In many cases, these methods are used in early detection of degradation and malfunction in reactor components. The neural networks are trained using the data obtained during normal fault free operation of the selected equipment or system. The faults are detected as the abnormal vibration pattern, generated from the equipment that is being monitored. The above-mentioned neural networks based concept has been applied in the fault detection of primary coolant pump shaft of the Experimental Breeder Reactor (EBR-II) [27]. Seker and colleagues [28] applied neural network model to detect damage to the bearing of induction motors in a Gas Cooled Reactor.

1.4.2. Application of Fuzzy Logic Systems in Reactor Environment

Fuzzy logic systems allow "approximate reasoning", by which an element can be assigned to a set with a degree of certainty. These systems try to imitate human reasoning that allows reasoning with uncertain facts to infer new facts, with a degree of certainty associated with each fact. The applications of fuzzy logic systems in nuclear reactors cover a wide range from traditional control applications like heater on/off, water level control etc. to advance applications like transient identification, signal prediction, etc. In general, the application of fuzzy logic in plant control application of nuclear reactor is to mimic the hardware based conventional controller, like Proportional Integral Derivative (PID) controller [29]. The fuzzy logic based controller has certain advantages over a PID controller. Fuzzy logic controllers are software based; therefore, they are more flexible and can be used for controlling both linear and non-linear systems.

A typical control application of fuzzy logic in nuclear reactors is in **power controlling**. Hah and Lee [30] applied a fuzzy logic based automatic power shape control mechanism for a Pressurized Water Reactor. Liu and colleagues [31] proposed a fuzzy logic controller tuned by the genetic algorithm in the power control system of Pressurized Water Reactor. In the proposed model, the genetic algorithm is used for tuning the membership functions of the fuzzy logic controller. Ramirez and colleagues [32] proposed an automated fuzzy based control scheme for tracking an optimal power profile in a Mexican research reactor.

Another application of fuzzy based controller reported is in the **water level control of steam generator**. During the low power operation of the reactors, poor control of water level of the steam generator can lead to emergency shutdowns. Therefore, the application of fuzzy logic controller in control of water in the steam generator has a scope of enhancing plant availability [33]. Kuan and colleagues [34] applied the fuzzy logic in the water level control of steam generator of a Pressurized Water Reactor. They used a set of linguistic rules based on the fuzzy logic, adopted from human operator experience, and achieved real-time control with improved controller performance. Similar concept was applied for Boiling Water Reactor by Lin and colleagues [35]. Habibiyan and colleagues [33] applied a fuzzy-gain-scheduled neural controller to control the steam generator water level of a Pressurized Water Reactor.

1.4.3. Application of Genetic Algorithms in Reactor Environment

Genetic Algorithms (GAs) are powerful and broadly applicable stochastic search and optimization methods, based on principles from evolution theory [36]. Most of the classical optimization methods generate a deterministic sequence of computation based on the gradient or higher-order derivative of objective function. GAs perform a multi-directional search by maintaining a population of potential solutions that make the search escape from local optima. The detailed study of the principles, operators and behavior of the algorithm is presented in Chapter 2. During the past few decades, genetic algorithms are being applied to solve many complex optimization problems inherent in different subsystems of nuclear reactors. The findings based on a literature survey conducted on the application of GA methods in the domain of nuclear reactor are presented below.

One of the applications of GA in the domain of nuclear reactors is in the **reactor power control.** Application of GA in reactor power control is focused in finding out the optimal way of using reactivity control mechanisms. In most of the cases, it is related to the selection and movement of control rods in the reactor core. GA models can help in finding the optimal control rod movements while satisfying all the safety criteria. A model based on GA for the power control of a Pressurized Water Reactor by simulating effective reactivity parameters was proposed by Marseguerra and Zio [37]. The optimization of power ascension path for a Boiling Water Reactor using GA was done by Lee and Lin [38]. Initially, they used a 3-D simulation code to model the core characteristics. Later, they improved the model by incorporating the neural network concepts for faster predictions of core characteristics [39]. In a recent study, Kim and colleagues [40] proposed the design of a load-following controller for Pressurized Water Reactor by using GA to optimize non-linear discrete speed of control rods.

Another well-investigated application of GA in the domain of nuclear reactors is in the **test interval optimization of safety systems.** The reactor safety systems are tested and maintained periodically in order to ensure their serviceability. The optimization of test intervals, based on risk (or unavailability) and cost (or resource expenditure) is a complex problem, with non-linear objective functions and constraints. Jiejuan and colleagues [41] studied the risk-cost models based on the GA for surveillance test interval and preventive maintenance period optimization of a Chinese Pressurized Water Reactor. Gopika and colleagues [42] applied GA for optimizing in-service inspection intervals of channel feeders of a Pressurized Heavy Water Reactor core. A GA model has been applied in the surveillance test policy of auxiliary feed-water system of a Pressurized Water Reactor by Pereira and Lapa [43]. A risk-cost model based on multi-objective genetic algorithm was applied to decide the surveillance test interval and the preventive maintenance period for an Indian Pressurized Water Reactor by Mishra and colleagues [44]. The study was later augmented by the use of clustering technique in order to improve the performance of the multi objective GA [45]. An advanced progressive, real-coded genetic algorithm was developed and applied for the availability-optimization of emergency core cooling system of two separate Pressurized Water Reactors, by Aghaie and colleagues [46]. Mehra and colleagues [47, 48] applied the GA based risk-cost optimization model, for surveillance test interval optimization of the passive standby decay heat removal system of Prototype Fast Breeder Reactor (PFBR).

Apart from the domains mentioned above, applications of GA are investigated in the various engineering design optimizations also. For example, Sacco and colleagues [49] applied GA in the **steam-turbine system** of reactors for finding the optimal fraction of mass flow rate to be extracted from each stage of the turbines of a Pressurized Water Reactor. The study conducted by the team revealed the robustness and efficiency of GA based model in finding possible changes of the existing plants' turbine extractions or in the design of new power plant turbine system. Kin and Moon [50] applied an optimization approach based on GA, for the **radiation shielding design** of a small-sized spaceship reactor (which acts as a power source for an unmanned spaceship).

Another important application of GA, i.e. GA in the **nuclear fuel management** has been taken up as the main focus of the work carried out in the thesis. The remaining part of the chapter presents a detailed description about the nuclear fuel management and the applications of intelligent optimization methods, including the GA. The findings presented are based on the literature survey carried out in the area of application of GA in the domain of nuclear fuel management.

1.5. NUCLEAR FUEL MANAGEMENT AND INTELLIGENT OPTIMIZATION METHODS

The prime aim of nuclear fuel management is to achieve higher fuel utilization during the reactor operation without compromising safety. Nuclear fuel management entails making decisions that influence how a nuclear reactor core's reactivity, neutronics flux, power, and burnup distribution vary in order to extract electrical energy in a cost effective way. The cost reduction is achieved through various means: maximization of fuel cycle length, maximization of the burnup, minimization of fuel inventory, and minimization of the inventory of reactivity control material. In essence, the common aim of nuclear fuel management problems is to achieve higher fuel utilization; but the objectives and constraints vary with the type of fuel management problem.

Generally, the nuclear fuel management problem has multiple objectives and constraints. When all these objectives and constraints are considered together, some of them will conflict with the other. Hence, any one of the final solutions represents some sort of compromise in which no further improvement in a given performance index can be obtained without a degradation in at least one of the other performance indices. Therefore, the goal of the nuclear fuel management is to identify the solution vector that suggests the best compromise among the objectives, while satisfying the given constraints.

1.5.1. General Classification of Nuclear Fuel Management Problems

Traditionally, nuclear fuel management has been divided into two categories, out-of-core and in-core fuel management. Out-of-core fuel management focuses on answering the questions "What to manufacture?" and "What to insert?" in the context of multi-cycle operations of the reactor. These questions are to be answered during the initial design stage of the reactor core. Out-of-core fuel management includes the decision on number and composition of fresh fuel assemblies. It also decides on the type and configuration of reactivity control materials. In essence, out-of-core fuel management addresses the design issues of the reactor core and strives to arrive at the optimal core design.

In-core nuclear fuel management focuses on answering the question "Where to position?" the fuel assemblies or reactivity control materials in the core. The in-core fuel management entails the arrangement of fresh and partially burned fuel assemblies and reactivity control mechanisms within the core that optimizes the performance of the reactor over the next operating cycle, while ensuring that operational constraints are always satisfied [51, 52]. The determination of location and orientation of different fuel subassemblies and reactivity control materials are the two issues normally addressed by in-core nuclear fuel management decisions. Majority of the nuclear fuel management problems where intelligent optimization methods like GA applied are coming under the in-core fuel management.

1.5.2. Nuclear Fuel Management Problems in Different Types of Reactors

The fuel management of Pressurized Water Reactor (PWR) and Boiling Water Reactor (BWR) are having many common features along with certain differences among them. In both types of reactors, the refueling is off-power i.e. refueling is done after shutting down the reactor. The refueling scheme is also similar in both the type of reactors, where shuffling of burned fuel assemblies is done. The use of burnable poisons as a reactivity control material is also common in PWR and BWR. In PWR, separate burnable poison rods which are similar to the fuel assemblies but manufactured with neutron absorbing materials, may also be present. Apart from the burnable poison, soluble poison like boric acid (called as chemical shim) is used for excess reactivity control in PWR. Since chemical shim is used to ensure reactor criticality at the desired core flow rate (known as criticality constraint), control rod positions and insertions are not coming as a decision variable in PWR in-core fuel management optimization problems. However, in the case of BWR, where chemical shim is not used, the criticality constraint is achieved by positioning of control rods. The positioning of control rods as a function of cycle burn up is called control rod program. In BWR, the in-core fuel management should consider optimum control rods program also, for each potential "loading pattern" of fuel assemblies which result in added complexity for the optimization problem [52]. The term "loading pattern" indicates the location and orientation of different subassemblies in the reactor core. The reactor physics based neutronics simulation codes used in BWR are usually 3-dimensional because of the presence of more number of fuel assemblies in the core, presence of strong axial heterogeneities and coolant voidity. This results in more computational time for in-core fuel management problems of BWR.

The in-core fuel management scheme of Pressurized Heavy Water Reactor (PHWR) is different from that of PWR and BWR. In the case of PWR and BWR, the fuel assemblies are loaded at the beginning of cycle and there is no fuel assembly movement until the reactor is shut down for refueling operations. There, the fuel management scheme aims to determine the best fuel arrangement throughout a fuel cycle. In PHWR, the refueling is on-power (i.e. while the reactor is in operation) and on a daily basis to continue the controlled chain reaction in the reactor and also to maintain the designed power distribution. Because of the on-power refueling scheme, the equilibrium core of the reactor is not uniquely defined. Generally, the time-average core calculation is used to search for an optimum power distribution by changing the number of fuel bundles loaded per refueling operation and adjusting the discharge burnups of the inner and outer cores in the PHWR.

The in-core fuel management scheme of the fast breeder reactor (FBR) is off-power which is similar to that of PWR and BWR. One major difference in FBR is that the reshuffling of burned fuel assemblies is not usually done during refueling operation. This is due to the bowing of subassemblies in response to fast neutron flux and temperature gradients. Another noteworthy point is that in FBR, neutron moderators are absent and hence neutron poison mechanisms (made up of neutron absorbing materials) like burnable poisons or chemical shims are also absent.

There are some special cases of in-core fuel management where instead of regular fuel reloading, some special cases of fuel loading patterns are considered as the objective. Examples of such special case fuel management problems are optimization of thorium loading in fresh core [53] and optimization of depleted uranium bundle loading in fresh core [54, 55] of PHWR.

1.5.3. Optimization Methods Applied in Nuclear Fuel Management

Optimal nuclear fuel management is a classical optimization problem in the field of nuclear engineering that was initially solved manually by experts. Finding out the optimal reactor core configuration is crucial for initial fuel assembly loading as well as periodic fuel assembly reloading, irrespective of the type of nuclear reactor. Therefore, the need for suitable optimization technique in this field was evident from the early stages of nuclear reactor design and implementation. All the early optimization methods applied in nuclear fuel management were using gradient search or hill climbing methods for finding the optimal solutions [56]. These methods were having the possibility of getting trapped in local optima. Therefore, the requirement of intelligent optimization methods for nuclear fuel management was realized from early stages of nuclear fuel management.

During the period of seventies and eighties, inventions of global optimization methods like GA by John Holland [57] and simulated annealing by Kirkpatrick et al. [58] became available. Nuclear fuel management problems found an immediate suitability for application of these global optimization methods [51]. During the past two decades, several other Computational Intelligence methods have also been used in nuclear fuel management field by many researchers. Some of them are: Tabu Search [59], Ant Colony Optimization (ACO) [60, 61], Ant-Q optimization [62], Particle Swarm Optimization (PSO) [63, 64], Artificial Bee Colony Optimization (ABCO) [65, 66], Harmony Search Algorithm (HSA) [67] and Continuous Firefly Algorithm (CFA) [68]. The methods listed above, come under the category of nature inspired intelligent algorithms. There are other types of optimization methods applied in the field, for example, Mixed Integer Programming [69], Estimation Distribution Algorithm (EDA) [70, 53] and Particle Collision Algorithm (PCA) [71]. Even though developments in other Computational Intelligence methods are taking place in parallel, GA still stands as a well-proven and widely accepted optimization technique in the field of nuclear fuel management. At present, many new variants of GA are being applied in nuclear fuel management of different types of reactors [72, 73, 74].

1.6. GENETIC ALGORITHMS IN NUCLEAR FUEL MANAGEMENT

One of the early applications of GA in nuclear fuel management was presented by Poon and Parks [51]. A detailed comparison between the GA and simulated annealing was presented in their work; the advantage of GA in the global search and GA's suitability for parallel computers was observed therein. The GA has got several interesting and intriguing applications in the domain of nuclear fuel management, as can be seen by the number of papers that have been published internationally. There are two major approaches in formulating nuclear fuel management optimization models using GA:

(i) Constrained optimization with penalty functions (referred to as Penalty-function GA in the rest of the thesis)

(ii) Multi-objective genetic algorithms (referred to as Multi-objective GA in the rest of the thesis)

In the case of as Penalty-function GA, the multi-objective optimization problem is converted in to a single objective by adding penalty functions and constraints. Multi-objective GA handles the multiple objectives altogether, by incorporating the concept of Pareto-optimality and dominance. The Penalty-function GA and Multi-objective GA are the methodologies studied in detail in the thesis, where in the applications of GA in nuclear fuel management are considered. Detailed description about these two methodologies is given in the next chapter (see Section 2.7).

There is yet another category, i.e. GA based on the parallel programs (referred to as Parallel GA) where the execution part of GA is divided to parallel executable units. The computational overheads due to complex neutronics calculations are involved in nuclear fuel management problems; hence, the parallel computing concepts of Parallel GA are promising. It is important to note here that the parallelization concept of Parallel GA is for dividing and distributing the computational burden among multiple processors. However, for handling the constraints of fuel management problems in Parallel GA, one has to follow the Penalty-function GA or the Multi-objective GA approach.

The findings from the literature survey carried out on the types of applications of GA in nuclear fuel management are summarized in Table 1.1. The table shows "where and how" different categories of GA are applied, in nuclear fuel management field. The term "Core design" in the table is used to refer the out-of-core fuel management problems. All other types of problems given in the table (indicated as "Loading pattern", "Burnable poisons" etc.) are coming under in-core fuel management.

Table 1.1: Summary of different types of GA in nuclear fuel management applications

Type of GA	Type of reactor	Type of problem	References
Penalty-function GA	Pressurized Water Reactor (PWR)	Loading pattern	70,74,75, 76,77,78
		Burnable poisons	79,80,81
		Loading pattern and Burnable poisons	51,82,83,84, 85,86,87,88
		Core Design	89,90
	Boiling Water Reactor (BWR)	Loading pattern	91,92,93
	Pressurized Heavy Water Reactor (PHWR)	Online Refueling	94
		Thorium Loading	53
	Advanced Gas cooled Reactor (AGR)	Loading pattern	95,96
	Research reactor	Loading pattern	97,98
Multi Objective GA	Pressurized Water Reactor (PWR)	Loading pattern and Burnable poisons	99
		Core Design	100
	Boiling Water Reactor (BWR)	Loading pattern and CRP	101,102,103
	Pressurized Heavy Water Reactor (PHWR)	Online Refueling	104
	Fast Breeder Reactor (FBR)	Loading pattern	105,106
	Research reactor	Loading pattern	107
Parallel GA	Pressurized Water Reactor (PWR)	Loading pattern	108
		Loading pattern and Burnable poisons	109
		Core Design	110,111

It is evident from the table that, the most commonly applied GA method in nuclear fuel management is the Penalty-function GA. In fact, the four articles listed in the table under Parallel GA also in turn follow the Penalty-function GA for handling the fuel management constraints. Hence, out of the forty one articles on nuclear fuel management based on the GA reviewed here, thirty two (including four articles coming under Parallel GA) are of the Penalty-function GA. The reason for this being the simplicity of this approach in GA model formulation, in which multiple constraints are converted in to a single objective function. The potential of Multi-objective GA in nuclear fuel management is not explored enough. Another important observation is that, even though there has been progress in the application of GA in nuclear fuel management of Pressurized Water Reactors and Boiling Water Reactors, similar applications in Pressurized Heavy Water Reactors and Fast Breeder Reactors were found to be limited.

1.7. MOTIVATION FOR THE RESEARCH

The review of literature indicated that there are only few studies that deal with application of GA in nuclear fuel management of Pressurized Heavy Water Reactors and Fast Breeder Reactors. Out of forty-one articles reviewed, three article pertain to PHWR and two relating to FBR (see Table 1.1). As these types of reactors are the main building blocks of the three-stage nuclear power programme of India (see Section 1.3), the task of developing GA based optimization procedures has been taken up. It is also envisaged that the study and application of different methodologies of GA will help in applying the intelligent optimization methods to more subsystems of these types of reactors.

From the survey of literature, it can be seen that the total optimization procedure is developed as a single module including that of neutronics simulation module. There is a scope for dividing the total optimization procedure in to: GA module, interface module and neutronics simulation module. The GA module is considered as the optimization module, the neutronics simulation module as the fitness evaluation module. An interface module is developed for communicating among these modules. Such configuration would offer the two major advantages: (i) allows developing more efficient communication mechanisms between the GA module and the neutronics simulation module (ii) minimization of the changes required when extending the application of optimization procedure to other studies of similar nature.

Further, it is seen from the literature survey that, the reason for opting for a particular GA methodology for a given optimization of the chosen parameters, is not addressed. To get a better understanding about the suitability and performance of two different methodologies, namely Penalty-function GA and the Multi-objective GA, can be applied and compared for a problem like optimization problems of nuclear fuel management.

1.8. OBJECTIVES AND SCOPE OF THE RESEARCH

As part of the study, different optimization methodologies based on the Genetic Algorithm are applied and evaluated in a set of diverse and less explored environments of Pressurized Heavy Water Reactors and Fast Breeder Reactors. The scope of the present work is mainly limited to the two methodologies of GA, i.e. Penalty function GA and Multi-objective GA in dealing with the nuclear fuel management problems. Apart from that, the scope of application of the standard procedure of GA in an engineering design optimization of the circulating water system of a reactor is also investigated.

The main objectives of the study can be summarized as follows:

- Application of GA in nuclear fuel management of Pressurized Heavy Water Reactors and Fast Breeder Reactors.

- Follow a modular approach in applying optimization methodologies based on the Genetic Algorithm in various subsystems of nuclear reactors.
- Study and verify the suitability of different methodologies of the Genetic Algorithm namely, Penalty-function GA and Multi-objective GA, for different types of nuclear fuel management optimizations.

1.9. ORGANIZATION OF THE THESIS

The thesis is organized in six chapters and the significant contribution in each of the chapter can be briefly summarized as follows:

- In the Chapter 1, an introduction to the research work is given. The description and findings from the literature survey carried out in the field of Computational Intelligence applicable in different nuclear subsystems, including the Genetic Algorithm applications in nuclear fuel management, are covered.
- Chapter 2 presents a study on GA and the evaluation of performance of the algorithm on a benchmark optimization problem. Operators and parameters of GA, used in the work carried out as part of the thesis are introduced in the chapter. The methodologies of GA applied in the optimization studies of nuclear fuel management are also introduced.
- Chapter 3 addresses the optimization study of engineering design conducted on steam condenser of Prototype Fast Breeder Reactor (PFBR).
- Chapter 4 explains the work carried out for the 220 MWe PHWR fuel bundle optimization.
- Chapter 5 covers the optimization studies carried out on two different core configurations of Fast Breeder Reactors.
- Chapter 6 summarizes the work carried out in the thesis, the conclusions drawn from the studies carried out and possible areas of future work.

CHAPTER 2

A STUDY ON GENETIC ALGORITHMS

In the chapter, the overall process and procedure of Genetic Algorithms, the experiments conducted on the performance of the Genetic Algorithm using a benchmark optimization function and the Genetic Algorithm methodologies followed in the nuclear fuel management studies are discussed. Various operators, parameters and methodologies of the Genetic Algorithms used in the work carried out as part of the thesis are also introduced in the chapter.

2.1. INTRODUCTION

Several biological concepts are applied in the domain of Computational Intelligence, and Genetic Algorithm (GA) is an example for that. GA is powerful and broadly applicable to several stochastic search and optimization processes, and is based on the principles derived from the evolution theory. The algorithm draws inspiration from Darwin's theory of the survival of the fittest and natural selection. GA was developed by John Holland [57] and later popularized by one of his students, David Goldberg, who successfully applied it to several engineering problems [112].

Not only the basic concepts, but also much of the vocabulary of GA has been borrowed from genetics and evolution theory. In GA, a part of the solution space that has been randomly chosen forms the 'population'. Then the population is iteratively improved using a set of genetic operations like, 'selection', 'crossover' and 'mutation'.

The GA population consists of a set of solutions each of which is known as ‘chromosome’ which forms the blueprint of an ‘individual’. GA chromosomes are made up of the basic units called ‘genes’; every gene normally controls the inheritance of a particular character in the solution space of the optimization problem. Any character of individuals can manifest itself differently; each different manifestation is said to be an ‘allele’ of the gene. Every iteration of GA simulates a ‘generation’ in natural evolution.

One important feature of GA is its ability to balance intelligently between exploiting the best solution and exploring the search space [113]. At the beginning of genetic search, there is a widely random and diverse population and crossover operator tends to perform widespread exploration of the search space. During the later stages of the search, the same crossover operator enables exploitation in the neighborhood of the solution space arrived. In addition, the genetic operators of the algorithm are designed as general-purpose (domain-independent) search methods; they perform essentially a blind search and could not always guarantee to yield an improved offspring.

2.2. GENETIC ALGORITHM: THE PROCESS FLOW

As stated in the introduction, GA is based on the principles of natural selection and natural genetics. The GA process mimics the “survival of the fittest” principle of the nature. The overall process of the standard GA is shown in the flowchart (Figure 2.1). The algorithm starts with generating an initial population of solutions. The population evolves through successive iterations of the algorithm, called generations. During each generation, the individuals (also known as chromosomes) of the population are evaluated, using some measures of ‘fitness’.

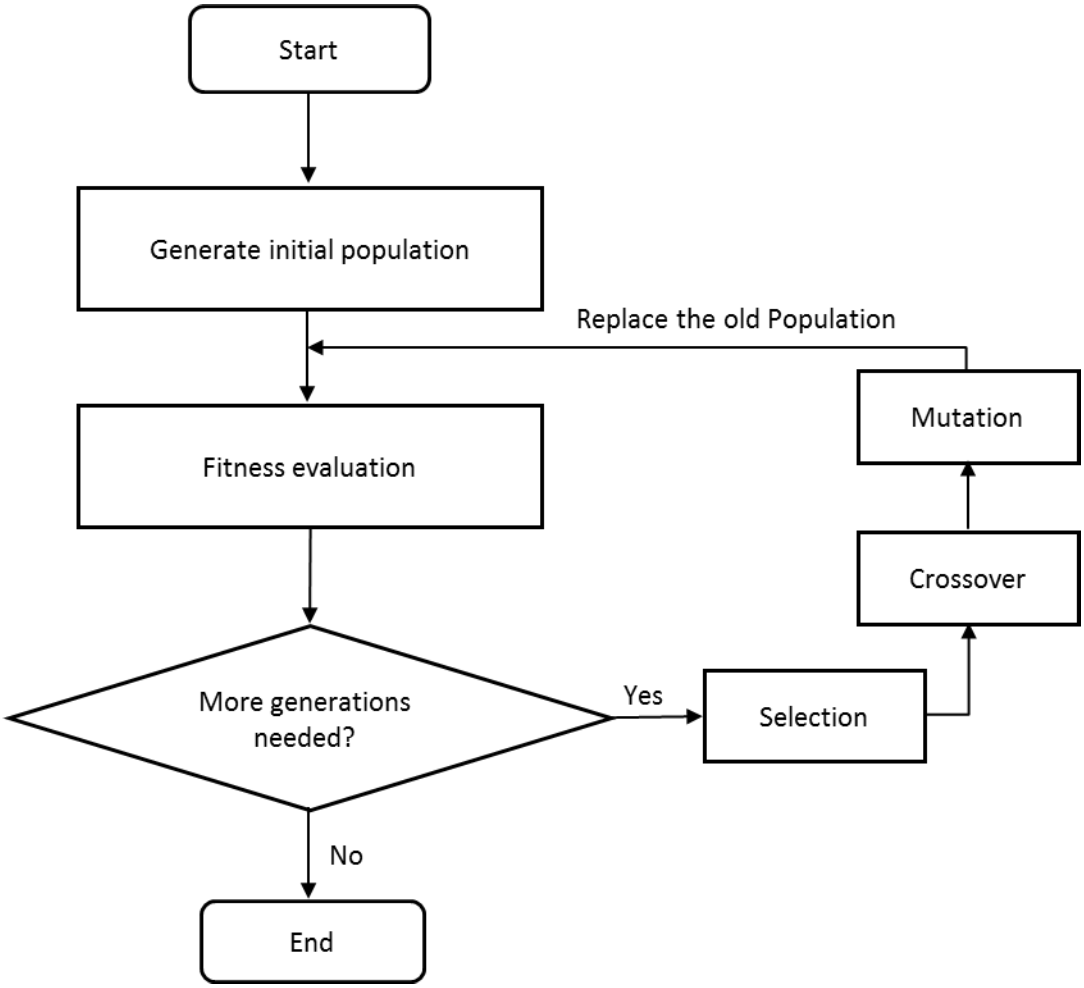


Figure 2.1: The process flow of standard Genetic Algorithm showing the major operations involved in the algorithm.

Based on the results of the evaluation, the parents are selected which participate in the 'reproduction' process and 'offspring' (or children) are generated. The basic genetic operators for the reproduction are crossover and mutation. Normally, these operations are happening with certain assigned probabilities. Therefore, occasionally there is a chance of copying of the parents without any changes to the offspring pool. In the standard GA procedure, the generation of offspring is repeated until a new generation of population is created. This refining process of the population is happening, till the termination condition (or conditions) specified in the algorithm is satisfied.

The ‘selection’ operation is carried out according to the fitness values assigned to the chromosomes; fitter chromosomes have higher probabilities of being selected. The ‘crossover’ operator acts on two chromosomes (or individuals) at a time and produces offspring by combining the two chromosomes’ features. The ‘mutation’ is a background operator which produces spontaneous random changes in various chromosomes. There are several representation schemes of chromosomes, suitable for different types of optimization problems. The types of operators are also varied accordingly. The representation of chromosomes is an important point that decides the overall performance of the algorithm. Next, the strategies adopted in representing the chromosomes are presented.

2.3. CHROMOSOME REPRESENTATION

Every search and optimization algorithm requires a suitable encoding scheme to represent the probable solutions. Traditionally, representation in GA is carried out using binary strings, mainly because of its amenability to simple implementation and ease of theoretical analysis. For several other real world applications, the binary encoding is difficult to apply, because the binary representation is not becoming a natural coding mechanism for such applications [36]. During the past two decades, various non-string representation methods have been created; one such scheme is real-number representation (also known as floating-point representation). The studies carried out as part of the thesis use binary representation and real-number representation. Therefore, those schemes are described in the following sections.

2.3.1. Binary Representation

In binary representation, every chromosome is a string of bits, 0 or 1. The concept is simple here; convert the entire decision variable of the problem to the suitable binary strings. The aggregation of such binary strings forms the chromosome.

The length of the binary chromosome depends on the number of decision variables, range of the decision variables and the required precision. Assume that the problem is with one decision variable of real number type. In which case, the required bits (denoted by 'm') can be calculated as:

$$2^{m-1} < (U - L) \times 10^{p} \ \leq \ 2^{m} - 1 \tag{2.1}$$

where, U and L represents the upper and lower limits of the decision variable respectively and 'p' represents the required precision or the required number of digits after the decimal point.

For example, let the range of the decision variable be between '-5' and '+5' and let the required precision be in four places after the decimal point. Substituting corresponding values in Eqn. 2.1, we get:

$$2^{m-1} < \big(5 - (-5)\big) \times 10^{4} \ \leq \ 2^{m} - 1$$

$$2^{m-1} < 10^{5} \ \leq \ 2^{m} - 1$$

$$2^{16} < 10^{5} \ \leq \ 2^{17} - 1$$

Therefore, the number of bits required for the example is 17. If the problem is with more than one decision variables, then for each of the decision variable, the length of the binary string is calculated separately according to the above rule.

The binary representation facilitates the use of several elegant genetic operators. The standard GA operators like crossover and mutation can perform efficiently on the binary strings. One disadvantage of binary representation is that, for large number of decision variables, the chromosome will become extremely lengthy. The operations on the lengthy binary chromosome become computationally time consuming and eventually affect the performance of the algorithm, especially when the problem space is large [113]. For example, for 100 variables with domain in the range (-500, 500) where a precision of six digits after the decimal point is required, the length of the chromosome becomes 3000 bits. This in turn, generates a search space of about 10^{1000}. For such a large search space, the binary representation performs poorly.

2.3.2. Real-number Representation

The real-number representation offers a natural way of chromosome representation without explicit encoding mechanism, when the decision variables are of real numbers. Each chromosome vector is coded as a floating-point vector, of the same length as the solution vector. Each element is forced to be within the desired range, and the operators like crossover and mutation are carefully designed to preserve this requirement [113]. The precision of such an approach depends on the underlying machine, but is generally much better than that of binary representation.

The real-number representation is more suitable for the continuous parameter optimization due to the encoding followed in the algorithm being better suited to the continuous problem domain [114, 115]. However, special genetic operators are required for them. The details about the operators suitable for real-number representation are discussed in the next section. The main objective of such operators is to move the solutions closer to the problem space. Therefore, in the case of real-number representation, the genetic operators for crossover and mutation are more specific to the type of problem.

In essence, it is difficult to conclude that any one particular representation is better in all the situations. Each among the two chromosome representation schemes is having advantages and some drawbacks. The choice between binary and real-number representation mainly depend on the problem domain. In order to understand the behavior of the two representation schemes, further studies are carried out as part of the thesis work (see Section 2.5).

Next, the important operations of the GA i.e. selection, crossover and mutation are discussed. When these three operations are carried out in coherence, the percentage of good individuals, in the population is getting improved generation wise.

2.4. GENETIC ALGORITHM OPERATORS

During each generation of GA, selection, crossover and mutation operations are sequentially applied to each individual by obeying certain probabilities of occurrence. They are the basic GA operators and several mechanisms are described in the literature, for implementing each among them [36, 112, 113]. The operators of GA employed in the work carried out as part of the thesis are discussed in detail below.

2.4.1. Selection Operation

As mentioned earlier, the selection of parents for the reproduction is according to the fitness values assigned to the chromosomes. The fitter chromosomes have higher probabilities of being selected. Selection is one of the main ways to maintain diversity in a population. The selection process can be handled in a number of ways. The "Roulette wheel" method (also known as fitness proportional selection) is the well-accepted selection method for several implementations of the GA [113] and the same is employed in the present work. The other type of selection method used in the present work is "Pareto-optimal front" based ranking which is used in multi-objective genetic algorithm implementations. The concepts related to the Pareto-optimal front are further discussed while dealing with the multi-objective genetic algorithm (referred to as Multi-objective GA). The Roulette wheel based selection procedure is detailed in the following section.

2.4.2. Roulette wheel Method

The Roulette wheel method defines a mechanism in assigning greater probability of being selected for the chromosomes with higher fitness values. In order to achieve this, the Roulette wheel can be thought of as marked with the proportionate fitness values of the chromosomes. As an example, let us consider there are five chromosomes with different fitness proportions (marked in percentage values) as shown in Figure 2.2. The chromosomes numbers are assigned from 1 to 5 in the figure. Let us assume that during every iteration, the wheel is rotated and the chromosome coming closer to the "selection point" is selected as the parent. As shown in the figure, the probability of selecting chromosome number 3 is the highest (i.e. 38%); next higher probability being that of chromosome number 1 (i.e. 31%), and so on. When the wheel is spun '*n*' times ('*n*' being sufficiently large number), the probability of individuals being selected for the reproduction will be in accordance with their fitness values. Therefore, the method improves the probability of the good individuals being selected, while keeping the diversity amongst the population. Once the individuals from the population are properly selected, they have to undergo the process of crossover and mutation. These operations are explained below.

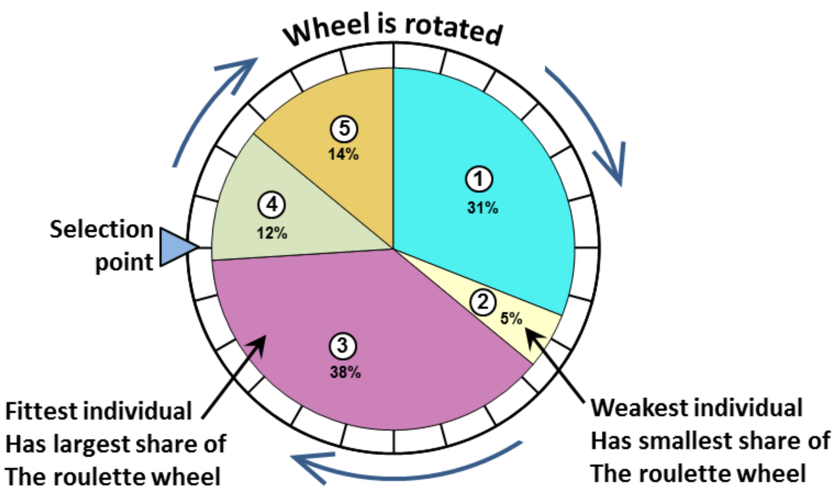


Figure 2.2: Illustration of Roulette wheel method. During each iteration of the selection procedure, the wheel is rotated and the individual coming closer to the "selection point" is selected.

2.4.3. Crossover

Crossover is the operation by which two parent chromosomes are combined to give two new chromosomes, which have some of the characteristics of the parents. During this process, there is a probability of getting offspring which are better than the parents. Crossover occurs during evolution according to a user definable crossover probability or crossover rate. The crossover rate defines a value that indicates how often the crossover operation is performed. The common type of crossover mechanism defined in literature, for binary representation, is the single-point crossover [112, 113]. In the case of real-number representation, special crossover methods like arithmetical crossover are to be used [113, 114]. The single-point crossover and the arithmetical crossover methods are further explained below.

2.4.4. Single-point Crossover

In single-point crossover, two binary chromosomes are randomly selected for the crossover operation. The chromosomes selected are cut once along the crossover point and the sections after the cuts are exchanged.

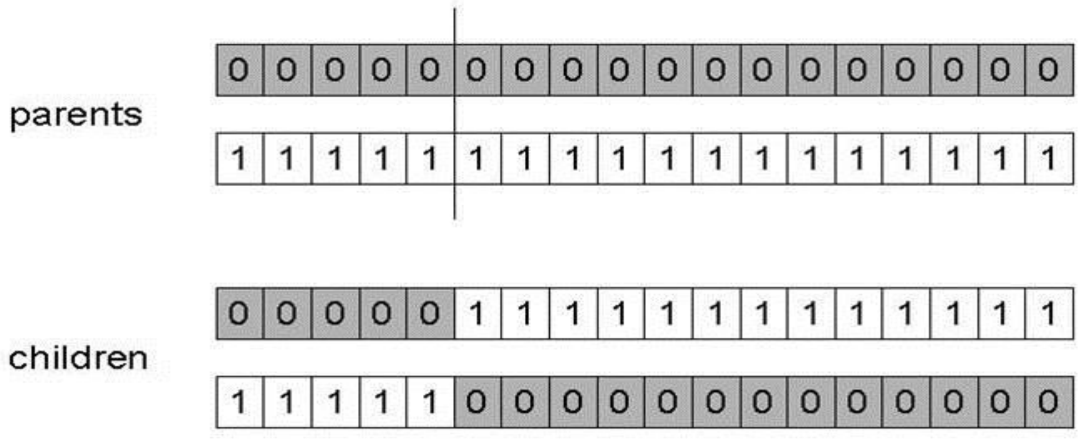


Figure 2.3: Illustration of the single-point crossover operation. The vertical line after fifth bit of 'parents' represents the crossover point. During the crossover, chromosome content is exchanged with respect to that point.

The crossover point is selected at random, along the length of the chromosome. In the example shown in Figure 2.3, the crossover point is selected just after the fifth bit of the 'parents' and is denoted by a vertical line. The resultant chromosomes (after the exchange of portions between the parents) are shown as 'children' in the figure. The operation results in the proper mixing of the genetic content of the parents during the production of the children.

2.4.5. Arithmetical Crossover

The Arithmetical crossover is a method suitable for real-number chromosome representation, which linearly combines two parent chromosome vectors to produce two new offspring [113]. Assume that x_1 and x_2 represent two real number type chromosomes. When the chromosome x_1 and x_2 are crossed, the resulting offspring are generated by the linear combination of the two vectors as:

$$ax_1 + (1 - a)x_2 \tag{2.2}$$

$$ax_2 + (1 - a)x_1 \tag{2.3}$$

where 'a' is a random real number, generated between the range 0 and 1.

As an example, consider two chromosome vectors, i.e. $x_1 = (10.05, 30.50)$ and $x_2 = (60.20, 15.80)$, the resulting offspring, o_1 and o_2, after arithmetical crossover operation is represented as:

$$o_1 = (a \times 10.05 + (1 - a) \times 60.20, \qquad a \times 30.50 + (1 - a) \times 15.80)$$

$$o_2 = (a \times 60.20 + (1 - a) \times 10.05, \qquad a \times 15.80 + (1 - a) \times 30.50)$$

At a particular instance, if a = 0.2672, then the resultant offspring chromosomes are:

$$o_1 = (46.80, 19.73)$$

$$o_2 = (23.45, 26.57)$$

(the results are represented after rounding off to two decimal places)

2.4.6. Mutation

Mutation adds new information in a random way to the genetic search process and ultimately helps the algorithm from getting trapped at local optima. It is an operator that introduces diversity in the population whenever the population tends to become homogeneous due to repeated use of the refining mechanisms of the algorithm. Mutation occurs during the evolution according to a user-definable mutation probability or mutation rate. This probability should usually be set fairly low. If it is set to high, the search will turn into a primitive random search. The common type of mutation mechanism described in the literature, for binary representation scheme is the flip-bit mutation [36]. In the case of real-number representation, special mutation methods like non-uniform mutation may have to be used. The flip-bit mutation and non-uniform mutation methods are used in different GA implementations in the present work. The two methods are further explained in the following sections.

2.4.7. Flit-bit Mutation

It is a mutation operator that simply inverts the value of the chosen bit (i.e., if chosen bit is 0, then it is inverted to 1, or vice versa). This mutation operator can only be used for binary representation. Figure 2.4 illustrates the flip-bit mutation method. In flip-bit mutation, one bit of the chromosome is selected randomly according to the given mutation rate. In the figure, the fourth bit of the 'parent' chromosome is undergoing bit inversion and the resultant chromosome is shown as the 'child'.

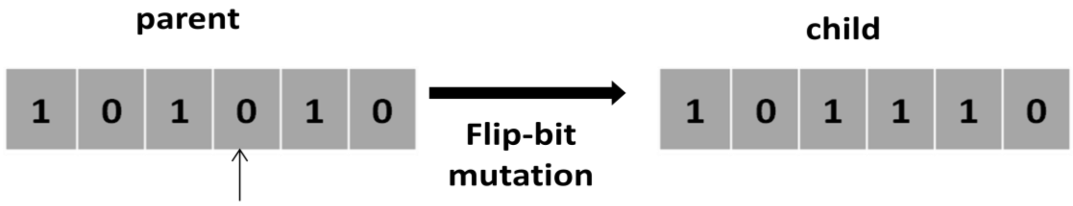


Figure 2.4: Illustration of the mutation operation. During the mutation, fourth bit of the 'parent' is inverted from '0' to '1', resulting into the 'child'.

2.4.8. Non-uniform Mutation

The mutation method applied for the real-number representation is the non-uniform mutation. A special feature of the method is that the mutation rate is not uniform i.e. it is related to the generation number of the algorithm [113]. The mutation rate decreases as the generation number increases and approaches a value close to zero. This property causes the algorithm to search the space uniformly initially and very locally at later stages and allows to fine tune the results.

If $x_1 = (v_1, \dots, v_{\mathrm{m}})$ is the chromosome vector, the element 'v_k' was selected for the mutation and the resultant chromosome vector is x_1'= $(v_1, \dots, v_k^l, \dots, v_m)$, then the non-uniform mutation operator is defined as:

$$v_k^l = \begin{cases} v_k + \Delta\,(t, UB - \ v_k) \text{ if a random digit is 0} \\ v_k - \Delta\,(t, v_k - \ LB) \text{ if a random digit is 1} \end{cases} \tag{2.4}$$

where, UB and LB are upper and lower bounds of 'v_k' and 't' is the generation number.

The function $\Delta\,(t, y)$ which returns a value in the range$[0, y]$ is defined as:

$$\Delta\,(t, y) = \ y\,(1 - \ r^{(1-\frac{t}{T})^b}) \tag{2.5}$$

where, 'r' is a random real number, generated between the range 0 and 1, 'T' is the maximal generation number of the genetic algorithm, and 'b' is a parameter determining the degree of dependency on the generation number. The value of 'b' is assigned to 5, for all the non-uniform mutation implementations throughout the work carried out as part of the thesis.

As a continuation of the example considered in Section 2.4.5, let us take the chromosome vectors created by the arithmetical crossover operation, i.e. $o_1 = (46.80, 19.73)$ and $o_2 = (23.45, 26.57)$. The chromosome vector (and also the particular element of that chromosome) for the mutation operation is selected randomly. Assume that one of the offspring (i.e. o_1) got from the crossover operation is undergoing the non-uniform mutation at 10th generation (i.e. t=10) of the algorithm. For this example, the other parameters like the maximal generation number of the genetic algorithm, (T) is taken as 50 and the generation dependency parameter, (b) is taken as 5. If the second element of 'o_1' is chosen for the mutation operation (i.e.$v_k = 26.57$). The UB and LB values represent the upper and lower bounds for the second element of the chromosome vector. Let UB and LB for the given example be 100.00 and 0.00 respectively. If the random digit generated as part of non-uniform mutation operator 'v_k^l' is 0, then according to the Eqn. 2.4,

$$v_k^l = 26.57 + \Delta\,(10, (100.00 - 26.57))$$

For a the random real number, $r = 0.60$, the $\Delta\,(t, y)$ function (see Eqn.2.5) results in 63.91 and the mutated element 'v_k^l' results in 90.48. In essence, for the example considered, the non-uniform mutation operation modifies the chromosome value from 26.57 to 90.48.

2.4.9. Elitism

According to the concept of 'elitism', the population is constructed by allowing some of the better chromosomes from the previous generation to carry over to the next, unaltered. This strategy ensures that the fitness values do not deteriorate as the generation moves along. This, relatively simple operation, improves the convergence of the algorithm manifold. The rate of convergence increases, as the number of elite members (i.e. the number of better individuals from the previous generations that replace an equal number of worse individuals) increases. However if the number of elite members is made too high, it can result in pre-mature convergence causing a lack of required diversity in the chromosome pool.

Some sort of elitism strategy is important for any type of genetic algorithms [116]. It makes sure that, a good solution found early on in the run can never be lost unless better solution is discovered. The elitism has been used in almost all the implementations of genetic algorithms, considered in the present work.

With an understanding of different encoding schemes and operators used in the present work, a look at the performance of these mechanisms on a typical sample problem is taken up and analyzed in the next section.

2.5. PERFORMANCE OF DIFFERENT SCHEMES AND OPERATORS OF GA

One of the aims of the present study is to get the understanding about arriving at the proper values of genetic parameters for the given optimization problem. The sample problem considered for the study is the Ackley's function. The choice of crossover rate and mutation rate is known to critically affect the behavior and performance of genetic algorithms. Even if there are general guidelines available in the literature about choosing typical values for crossover and mutation rates, it is desirable to carry out trial-and-error experiments because the optimal values of those rates are specific to the problem under consideration [117]. The insight got from this study is utilized in selecting the genetic parameters for the works related to the applications of GA in various subsystems of reactors, mentioned in the subsequent chapters.

2.5.1. Ackley's function

Ackley's function is a continuous and multimodal (with multiple optima) test function, widely used for testing optimization algorithms, obtained by modulating an exponential function with a cosine wave of moderate amplitude [36]. The function has multiple local minima and single global minimum, at the origin with function value 0. The topology of the Ackley's function is shown in Figure 2.5.

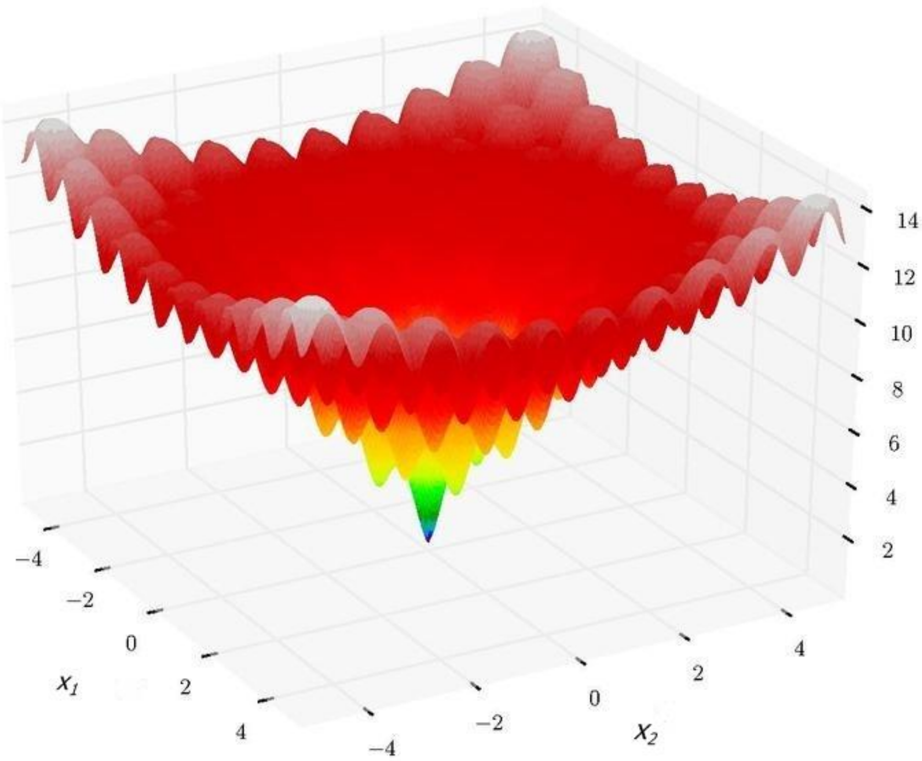


Figure 2.5: Ackley's function. The function is characterized by multiple local minima and single global minimum (global minimum value=0) at the center (source: https://en.wikipedia.org/wiki/Test_functions_for_optimization)

The Ackley's function is given as:

$\min f(x_1, x_2) =$

$$-c_1.exp\left(c_2\sqrt{\frac{1}{2}}\sum_{j=1}^{2} x_j^2\right) - \ exp\left(\frac{1}{2}\sum_{j=1}^{2} cos(c_3.x_j)\right) + c1 + e \qquad (2.6)$$

where, c_1=20, c_2=0.2, c_3 = 2π, e = 2.71282 and domain of the problem is :

$$-5 \ \leq x_j \leq 5, \ \ j = 1{,}2$$

The Ackley's function provides reasonable test cases for genetic algorithm search, since it causes moderate complications to the search [36] by having multiple local minima. A strictly local optimization algorithm that performs hill climbing would surely get trapped in local optimum, but a search strategy that scans a slightly bigger neighborhood would be able to cross the intervening valleys toward increasingly better optima. Therefore, the Ackley's function is selected for studying the performance of the algorithm. In the study, the function is used to test the performance of various representation schemes and operators of the standard GA.

2.5.2. General Points about the Study

One of the common methods of assessing the performance of GA is by measuring the average fitness value in each generation [36, 113]. The average fitness is calculated by adding the individual fitness values obtained from the fitness evaluation step of the algorithm, and dividing it by the population size. Since the present study is with the objective of minimization, the lower values of average fitness are better. The best possible average fitness value is zero for the Ackley's function. The two representation schemes implemented in the study are binary and real-number representations. The implementations are carried out using 'C' programming language. The experiments are conducted on a computer system with Intel Xeon dual six-core CPU@ 3.06 GHz and 48 GB RAM. The chromosome length for binary representation is of 50 bits; the first 25 bits to code the variable x_1 and the next 25 bits to code variable x_2 of the Ackley's function (see Eqn. 2.6). In the case of real-number representation, one chromosome is represented by two double precision real numbers; the first number represents variable x_1 and the second one represents variable x_2. Throughout the remaining discussion, 'CR' represents the crossover rate, 'MR' represents the mutation rate, and 'EL' represents the number of elite members. The algorithm parameters set in this experiment are listed below:

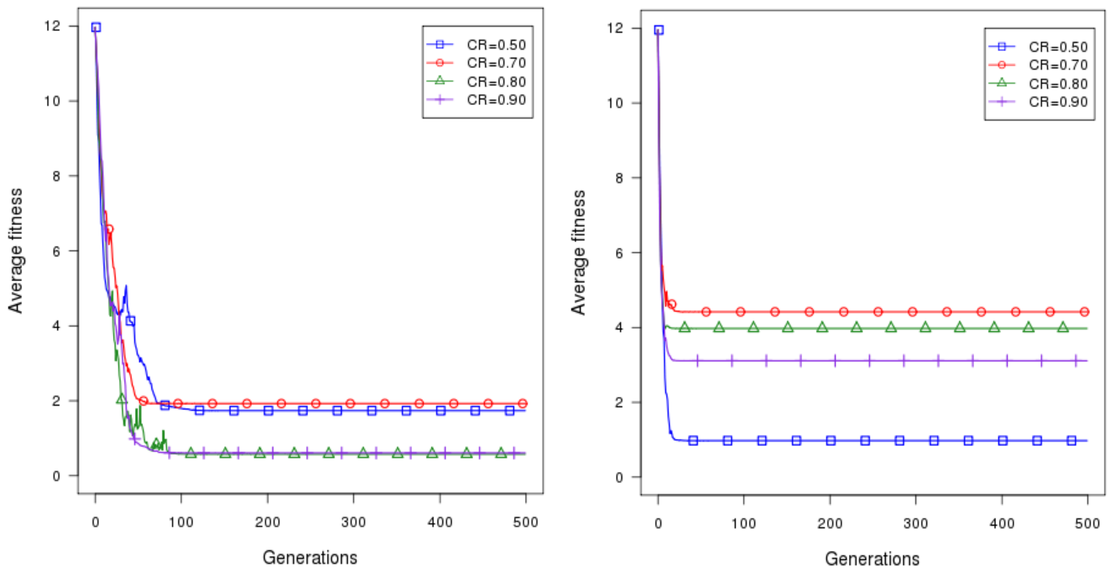
Crossover	:	Single-point (for binary representation),
		Arithmetic (for real-number representation)
Mutation	:	Flip-bit (for binary representation)
		Non-uniform (for real-number representation)

The parameters set common in the GA implementations of the two representation schemes are:

Selection	:	Roulette wheel
Population size	:	100
Generations	:	500

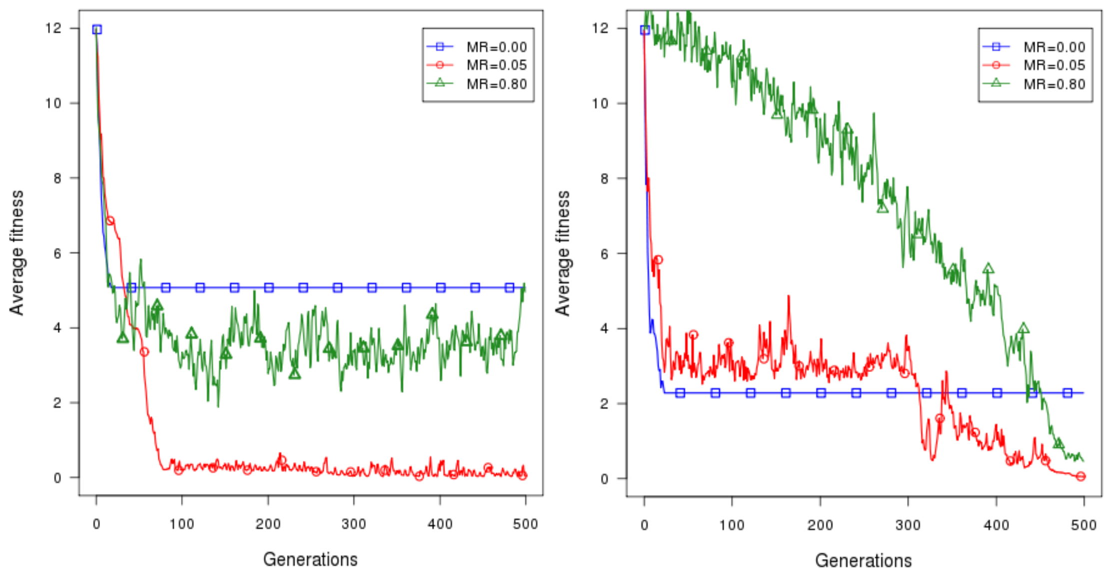
2.5.3. Effect of Variations in Crossover and Mutation Rates

The aim of the present study is to understand how the GA is getting influenced by different crossover and mutation rates. At first, let us consider solving the Ackley's function using GA, by changing the crossover rates, in the absence of other genetic operators. The cross over rate controls the capability of GA in exploiting a located hill of the search space to reach the local optima. The general guidelines available in the literature suggest that the typical values of crossover rate (CR) are above 0.5, i.e. crossover probability are above 50% [112, 113, 117]. As part of the present study, several trial runs have been conducted by varying the crossover rate (for both binary and real-number representations) in the range between 0.5 and 0.9. The mutation rate (MR) and the number of elite members (EL) are assigned to zeroes i.e., MR=0 and EL=0. The results of the trial runs are shown in the Figure 2.6. The effect of crossover rate on the binary representation of GA is shown in the Figure 2.6(a) and that on the real-number representation is shown in Figure 2.6(b). As shown in the figure, the crossover operation has more visible impact for the binary representation than the real-number representation (in the absence of other genetic operators). The reason for this can be attributed to the arithmetic crossover mechanism implemented for the real-number representation, where the exchange of genetic content is less vigorous in the absence of the mutation operation.



(a) (b)

Figure 2.6: The results with different crossover rates, in the absence of mutation and elitism. (a) Binary representation (b) Real-number representation.



(a) (b)

Figure 2.7: The results with different mutation rates (in the absence of crossover and elitism), i.e. absence of mutation (MR=0.00), typical small value (MR=0.05) and very high value (i.e. MR=0.80). (a) Binary representation (b) Real-number representation.

Another point is that, there is no uniform improvement in arriving better solution, when the crossover rate is increased from 0.5 to 0.9. This point is valid for the binary as well as the real-number representations.

The influence of mutation rete on the standard GA is studied next. The mutation operator helps the GA from being trapped in local optima of the entire search space. The mutation rate (MR) controls the ability of the algorithm in exploiting new areas of the search space. According to the general guidelines available in the literature, smaller values of mutation rates (when compared with the crossover rates) are commonly adopted [112, 113, 117]. The mutation rates considered for the studies are in the range between 0.00 and 0.80. During the trial runs, the other genetic operators (i.e. crossover and mutation) are made absent, by assigning the crossover rate and the number of elite members to zeroes i.e., CR=0 and EL=0. The effect of crossover rate on the binary representation of GA is shown in the Figure 2.7(a) and that on the real-number representation is shown in Figure 2.7(b). By conducting different trial runs, the absence of mutation (MR=0), a typical small value of mutation rate (MR=0.05) and a high mutation rate (MR=0.80), are compared. As can be observed from the figures, the absence of mutation as well as the high mutation rates are not supporting the fast convergence of the algorithm. Therefore, it can be concluded that a suitably selected small value of mutation rate can improve the convergence of the algorithm.

The effect of the GA operators on the performance of the algorithm, by considering one operator at a time, is illustrated above. The aggregate performance of the algorithm is analyzed next, by considering the algorithm operators altogether.

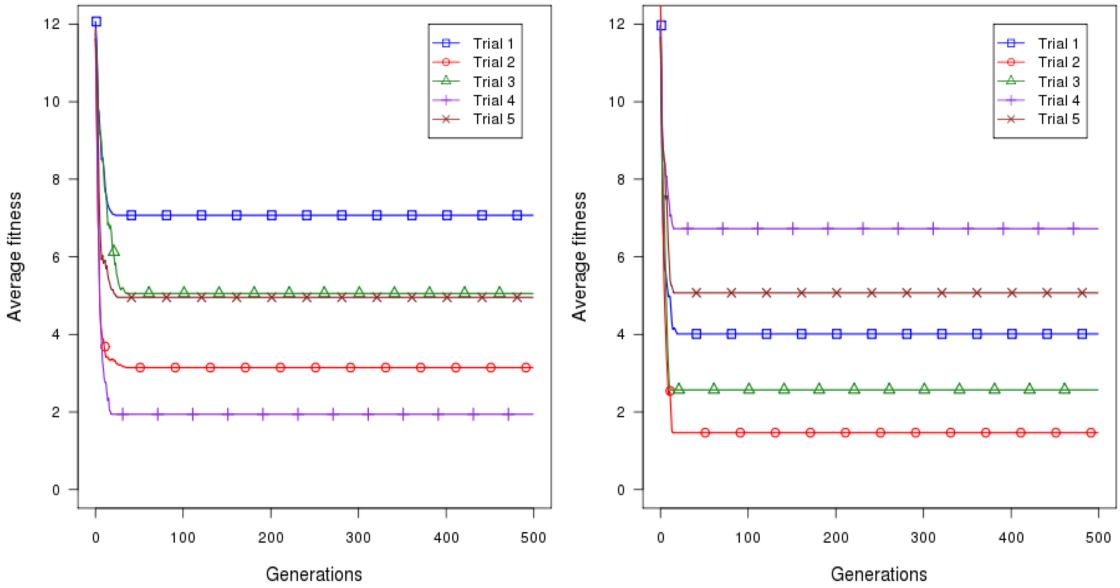
2.5.4. Aggregate Performance of the Algorithm

The aggregate performance of the GA indicates the convergence of the algorithm, when suitable values of genetic parameters are assigned for the algorithm operators. The suitable values are arrived at based on the initial trial runs carried out by varying the genetic parameters, like crossover rate, mutation rate and the number of elite members. In order to get a clear understanding of the influence of these parameters altogether, fifty trial runs are carried out (for both the binary and real-number representations) in the absence of crossover, mutation and elitism (i.e. CR=0, MR=0, EL=0). The results of five trail runs (selected randomly from the fifty trial runs) carried out in the absence of crossover, mutation and elitism are shown in Figure 2.8. The results of binary representation are shown in Figure 2.8(a) and the results of real-number representation are shown in Figure 2.8(b). Similarly, the results of five trail runs (selected randomly from the fifty trial runs) carried out by assigning suitable values to the genetic parameters are shown in Figure 2.9. The parameter values assigned for binary as well as real-number representations are: CR=0.80, MR=0.05 and EL=5. The results of binary representation are shown in Figure 2.9(a) and the results of real-number representation are shown in Figure 2.9(b). It can be observed form the figures that, assigning suitable values to the genetic parameters are important for the convergence of the algorithm.

The aggregate performance of the algorithm is further investigated by considering consolidated results of fifty trial runs. The parameters considered for measuring the performance of the algorithm are listed as:

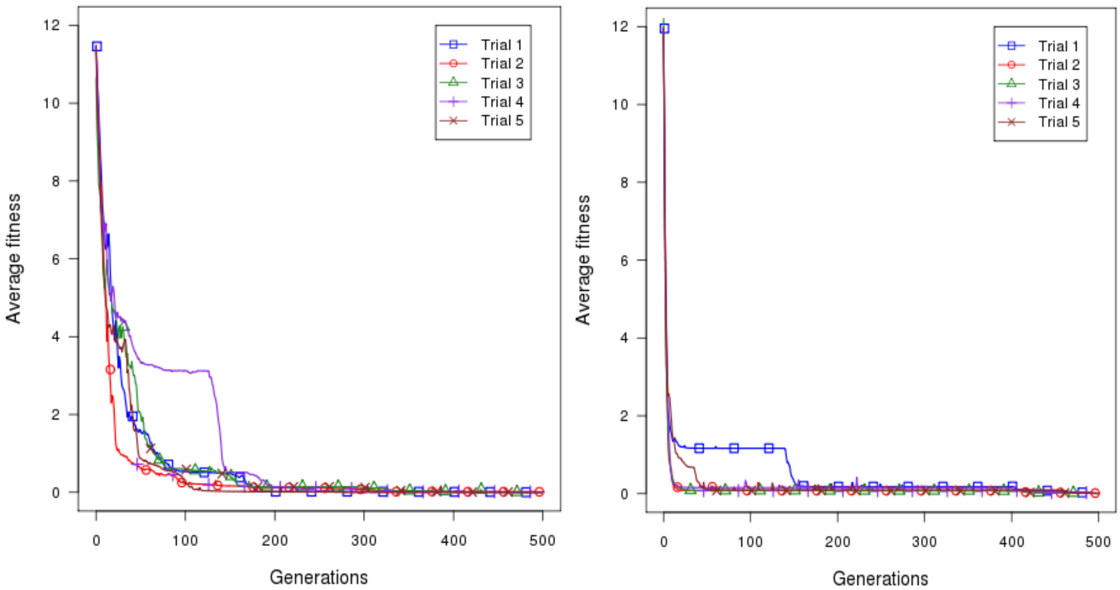
Best value: For every trial run, the best solution among the population (100 is the population size) produced in the final generation is recorded. The “Best value” denotes the best solution obtained amongst the 50 trial runs results.

Mean value: Similar to the case of Best value calculation, the best solution among the 100 solutions of the population produced in the final generation is recorded.



(a) (b)

Figure 2.8: The results of the trial runs (five results selected randomly from the fifty trial runs) in the absence of crossover, mutation and elitism (i.e. CR=0, MR=0, EL=0). (a) Binary representation (b) Real-number representation.



(a) (b)

Figure 2.9: The results of the trial runs (five results selected randomly from the fifty trial runs) with CR=0.80, MR=0.05 and EL=5. (a) Binary representation (b) Real-number representation.

The arithmetic mean of the Best values obtained for the 50 trial runs is denoted as "Mean value".

Standard deviation: This value represents the Standard deviation of the "Mean value" over the 50 trial runs.

The result of the experiment is furnished in Table 2.1. Different sets of fifty trial runs are conducted with elitism (EL=5) and also without elitism (EL=0) and the consolidated results are furnished in the table. During the experiment the crossover and mutation rates are assigned as CR=0.80 and MR=0.05.

Following observations can be made from the results given in the table:

- The algorithm is able to converge to "near zero" values for both binary and real-number representations in the presence of elitism (EL=5).
- In the absence of elitism (EL=0), the performance of binary representation is better than real-number representation.
- The presence of elitism improves the stability and robustness of the algorithm, which is evident from the lower values of mean and standard deviation obtained with the presence of elitism.

Table 2.1: The results of the aggregate performance of the algorithm by conducting fifty trial runs, in the presence and absence of elitism (with CR=0.80 and MR=0.05)

	Binary representation		Real-number representation	
	EL=0	EL=5	EL=0	EL=5
Best value	0.0004	0.0003	0.0038	0.0007
Mean value	0.0451	0.0081	0.3255	0.0120
Standard deviation	0.0649	0.0188	0.2601	0.0085

The study conducted on the performance of various representation schemes and algorithm operators has given the understanding on how the algorithm performance is influenced by them. The inferences from the results of the experiments help in selecting the suitable operators and the parameter values when GA is applied in solving the optimization problems related to different nuclear reactor subsystems. Nuclear fuel management is one such application considered in the present work. The overall procedure typically followed when genetic algorithms are applied in the nuclear fuel management is described in the following section.

2.6. GENETIC ALGORITHM IN NUCLEAR FUEL MANAGEMENT

The nuclear fuel management problems, their general classification and applications in various reactor subsystems are described in Chapter 1. When compared with the implementation of Ackley's function, the optimization problems of nuclear fuel management are more complex for mathematical formulation and algorithm implementation. While the Ackley's function has only one objective, typically the fuel management problems are with multiple objectives and constraints. Therefore, the standard GA that was considered for the implementation of the Ackley's function, cannot directly be applied to majority of the nuclear fuel management problems. However, some of the representation schemes and the operators available for the standard GA can be adopted, while formulating the GA methodologies for nuclear fuel management studies.

The overall procedure followed in the development of GA for nuclear fuel management problems, considering their special features and requirements, is considered next. The optimization procedure includes GA module (that includes steps of the standard GA procedure), interface module, and neutronics simulation codes, as shown in the flowchart of Figure 2.12. The fitness evaluation function of fuel management problem involves calling neutronics simulation codes which require complex mathematical calculations and take more computational time.

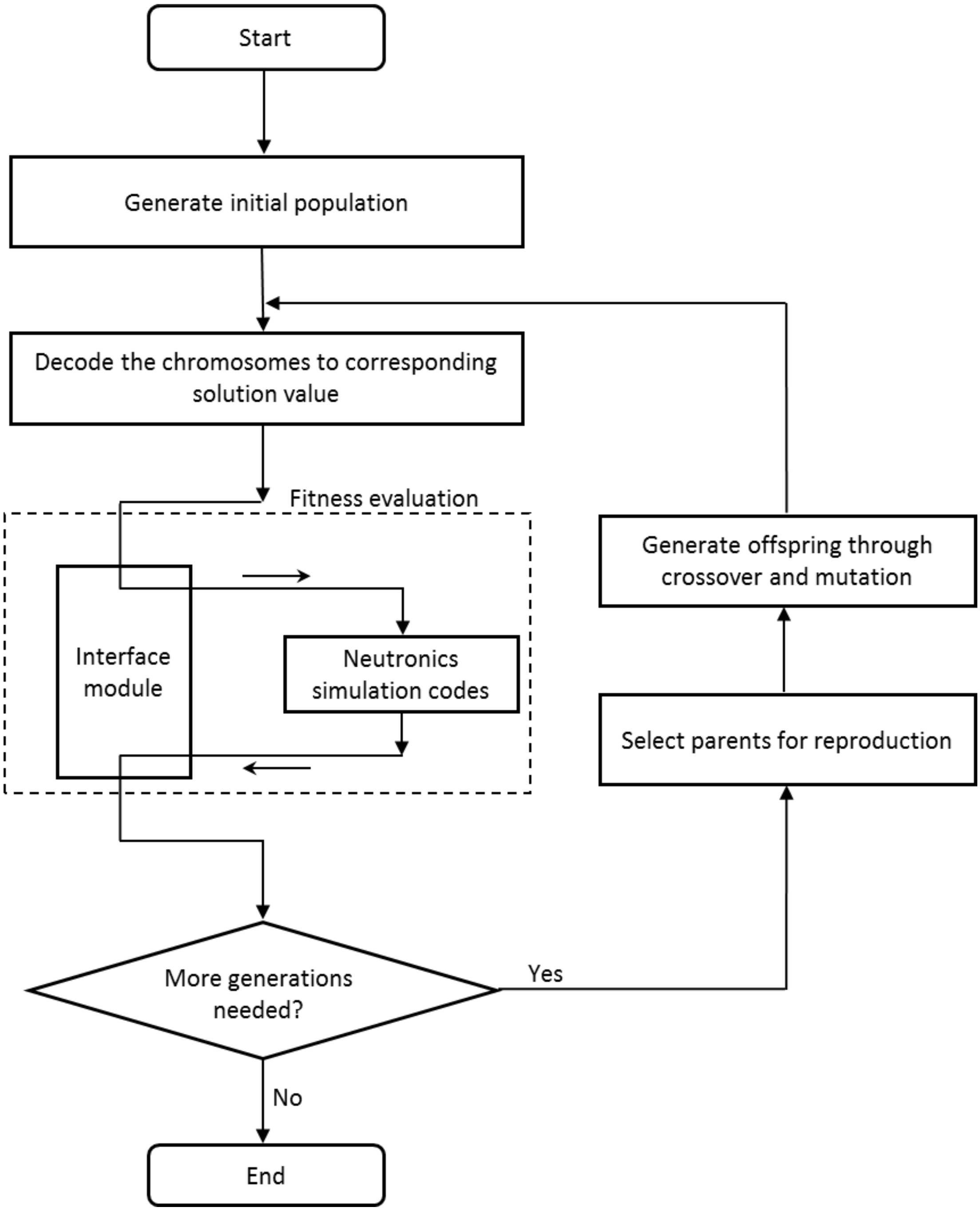


Figure 2.10: Overall flowchart of GA based optimization procedure that is commonly followed in the nuclear fuel management problems

The understanding about the basic functionalities and the overall working of neutronics simulation codes are necessary, for the development the optimization procedure. This can be achieved by conducting several independent trial runs of the neutronics simulation codes. The trial runs are conducted by proving typical input test conditions to the codes and verifying the output generated. Conducting such trial runs gives the understanding about the locations of important parameter values within the input and the output files of the neutronics simulation codes. During the optimization procedure, the input files are to be modified and output files are to be read back, in an automated way i.e. without user intervention. Therefore, the experience gained from conducting such trial runs is useful for the development the optimization procedure.

As can be seen in the flowchart, there is an interface module present between the GA module and the neutronics simulation codes. Most of the neutronics simulation codes used in the nuclear fuel management have been developed in FORTRAN programming language and are specific to the type of the reactor [51, 81]. If the GA implementation part is developed in any other language (for example, ‘C’ programming language is used in the present work), then the interface module should be able to generate the input files (without any user intervention) which satisfy the requirements of the neutronics simulation codes. Similarly, the required output values generated by the neutronics simulation codes should be searched and read by the interface module and given back to the GA module for further calculations. The step mentioned above involves searching of patterns in big output files generated by the neutronics simulation codes. Therefore, efficient pattern searching mechanisms are to be incorporated in the interface module. Another important functionality of the interface module is to carry out the whole procedure in an automated way.

A modular approach has been followed in the present work, for the formulation of the optimization procedure of nuclear fuel management problems, which facilitates in extending the applications easily to other reactor cores. Conceptually, the GA implementation part share many common features among the fuel management optimization of different reactor cores. Similarly, the algorithm methodologies (like Penalty-function GA and Multi-objective GA) are also commonly applicable for different nuclear subsystems. By considering these factors, the development of interface module is carried out in the present work. In essence, the interface module acts as a decoupling mechanism that separate the implementation details of the GA from the neutronics simulation codes.

In the above discussion, the overall GA based optimization procedure of the nuclear fuel management has been dealt with. Next, the different approaches in formulating nuclear fuel optimization model for GA, is discussed.

2.7. THE TWO GA METHODOLOGIES APPLIED : PENALTY-FUNCTION GA AND MULTI-OBJECTIVE GA

The literature survey carried out as the part of the present work indicates that, the major approaches in formulating GA based nuclear fuel optimization models are penalty functions based approach and multi-objective optimization approach. The corresponding two algorithm methodologies applied in various reactor subsystems, as part of the present work, are:

(i) Penalty functions based Genetic Algorithms (referred to as Penalty-function GA in the rest of the thesis)

(ii) Multi-objective Genetic Algorithms (referred to as Multi-objective GA in the rest of the thesis)

These two methodologies are further elaborated in the following sections.

2.7.1. Penalty-function GA

In the case of Penalty-function GA, the multi-objective problem of fuel management optimization is converted into single objective by adding penalty functions and constraints. In general, this approach transforms a constrained optimization problem to an unconstrained optimization problem by defining a suitable penalty function. The penalty function is needed to be formulated in such a way that, it should not affect the actual objective function when constraints are not violated. On the other hand, in the case of constraint violation, the penalty function will decrement the value of objective function, in accordance with the degree of constraint violation [113, 118]. If the penalty coefficients and the constraints are properly selected, this approach will give feasible solutions. Therefore, the proper selection is important for the convergence of Penalty-function GA. Generally, the selection of the penalty coefficients and the constraints are based on the inputs from the reactor core designer.

The application of the Penalty-function GA in the nuclear fuel management field is explained by taking a typical example from the optimization of a fuel assembly loading pattern (the term "loading pattern" indicates the location and orientation of different subassemblies in the reactor core). In the Chapter 1, we have seen that the Penalty-function GA is applied in many such applications (see Table 1.1). Generally, in the loading pattern optimization problem, the major objectives are to find the optimal loading pattern that maximizes the effective multiplication factor (K_{eff}) and to minimize the power peaking factor (*PPF*) [97]. The maximization of multiplication factor can help in extending the cycle length of the refueling operation. In addition, the power peaking factor should be as low as possible, because it defines the highest local power density of the fuel subassemblies. These objectives are considered as competing and conflicting with each other, because any attempt at maximization of the effective multiplication factor (K_{eff}) is limited by the power peaking factor (*PPF*) constraint. In summary, the loading pattern optimization considered in the example is a bi-objective optimization problem with conflicting objectives. The penalized objective function can be formulated according to the penalty function method as follows:

$$Max\ (Fitness) = K_{eff} - P1, \tag{2.7}$$

where, *Max* represents maximization, *Fitness* represents the value used by the genetic algorithm for the fitness evaluation of the solution, and *P*1 represents the penalty function. The penalty function (*P*1) is further defined as:

$$P1 = (PPF - PPF^{0}) \times Ap, \tag{2.8}$$

where, PPF^0 represents the maximum permitted limit for the power peaking factor, and *Ap* denote the penalty coefficient selected to give proper weightage to the penalty function. For normal solutions, when $PPF < PPF^0$, the value of *P*1 becomes negative and is added to the *Fitness* (see Eqn.2.7). During constraint violations, when $PPF > PPF^0$, the value of *P*1 becomes positive and is subtracted from the *Fitness*. This way, the penalty function improves the fitness values for better solutions with smaller *PPF* and at the same time penalizes (decrease) the fitness values for constraint violations.

In the above example, the proper selection of penalty coefficients and the constraints yield feasible solutions. One issue with the Penalty-function model is that, the multiple objectives are converted to single objective, which leads to the identification of solutions having less diversity in the trade-off surface. Another problem of the approach is that it is not always easy to aggregate these multiple objectives into a single performance index, until their relative importance is well understood. However, instances of the application of the Penalty-function GA are comparatively more in number for nuclear fuel management optimizations, mainly because it is easier to model and implement. In the case of fuel management applications, where the diversity in the solutions are of less importance and the savings in the time of the algorithm implementation is of more importance, the Penalty-function GA becomes a better choice.

2.7.2. Multi-objective GA

Multi-objective GA relies on the concepts of Pareto-optimality and dominance. Essentially, the main task of the Multi-objective GA is to find the Pareto-optimal solutions for the given problem with multiple conflicting objectives [116,119]. The Pareto-optimal solution is the one in which an improvement in one of the objectives requires a degradation of another. The set that consists of all the Pareto-optimal solutions for a given problem forms the Pareto-optimal front (or non-dominated front). The method makes it possible to identify the "trade-offs" between conflicting objectives in a single optimization run. The best solution is subjective and depends on the need of the designer or the decision maker. The multi-objective optimization concepts followed in the Multi-objective GA can formally be stated as follows [119]:

Find the vector, $\bar{x}^* = [x^*_1, x^*_2, \ldots, x^*_n]^T$ of decision variables which will satisfy the 'm' inequality constraints, i.e.,

$$g_i(\vec{x}) \geq 0, \; i = 1,2,\ldots, m, \tag{2.9}$$

and the 'p' equality constraints, i.e.,

$$h_i(\vec{x}) = 0, \; i = 1,2,\ldots, p, \tag{2.10}$$

and optimize the vector function, i.e.,

$$\bar{f}(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}), \ldots, f_k(\bar{x})]^T \tag{2.11}$$

The constraints given in Eqns. (2.9) and (2.10) define the feasible region 'β', which contains all the admissible solutions. The vector $\bar{x}^*$ denotes an optimal solution in 'β'. In the context of multi-objective optimization, it is difficult to arrive at a situation where a single vector $\vec{x}^*$ represents the optimum solution to all the objective functions.

The concept of Pareto-optimality comes handy in the domain of multi-objective optimization. A formal definition of Pareto-optimality from the viewpoint of the minimization problem can be given as follows:

A decision vector $\bar{x}^*$ is called Pareto-optimal, if and only if there is no $\bar{x}$ that dominates, $\bar{x}^*$ i.e., there is no $\bar{x}$ such that

$$\forall i \in 1,2,\ldots.,k, f_i(\bar{x}) \leq f_i(\bar{x}^*) \qquad (2.12)$$

and

$$\exists i \in 1,2,\ldots..k, f_i(\bar{x}) \leq f_i(\bar{x}^*) \qquad (2.13)$$

In other words, $\bar{x}^*$ is Pareto-optimal, if there exists no feasible vector $\bar{x}$ which causes a reduction in some criterion without a simultaneous increase in at least one other.

The concept of Pareto-optimality and non-domination is further illustrated in Figure 2.13. In the illustration, two objectives, f_1 and f_2 are minimized simultaneously. The shaded region represents the complete set β of feasible solutions. The 'x_i's (i.e. $x_1, x_2, \ldots x_n$) represent solutions in the objective space. The solutions, x_1, x_2, x_3, x_4, x_5 and x_6 are not dominated by any other solution in β. Hence, the set $\{x_i\}, i = 1,2, \ldots ,6,$ represents the Pareto-optimal set.

A multi-objective optimization approach should achieve the following three conflicting goals [119]:

(i) The best-known Pareto front should be as close as possible to the true Pareto front. Ideally, the best-known Pareto set should be a subset of the Pareto-optimal set.

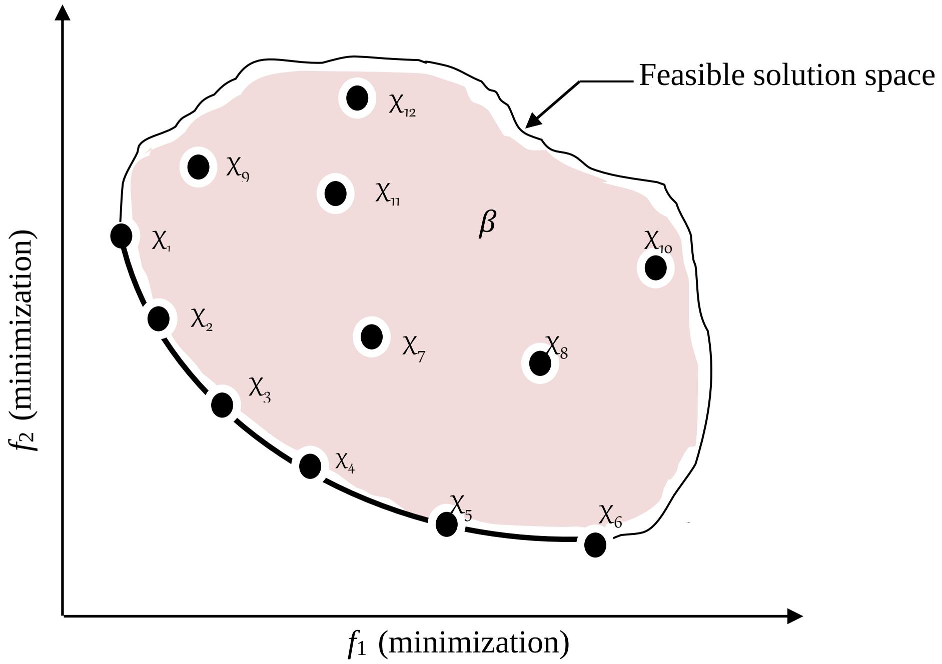


Figure 2.11: Illustrative example of Pareto-optimality and non-domination

(ii) Solutions in the best-known Pareto set should be uniformly distributed and diverse over the Pareto front, in order to provide the decision maker a true picture of tradeoffs.

(iii) The best-known Pareto front should capture the whole spectrum to the Pareto front. This requires investigating solutions at the extreme ends of the objective function space.

For a given computational time limit, the first goal is best served by focusing (intensifying) the search on a particular region of the Pareto front. In contrast, the second goal demands the search effort to be uniformly distributed over the Pareto front. The third goal aims at extending the Pareto front at both ends, exploring new extreme solutions.

2.7.3. Different Types of Multi-objective GA

During the last two decades, a number of different flavours of the Multi-objective GA are evolved and applied to solve several real-world optimization problems. There are two basic categories of the Multi-objective GA, namely "non-elitist Multi-objective GA" and "elitist Multi-objective GA" [116,120]. According to the concept of 'elitism', a fixed number of GA chromosomes having higher fitness values are considered as elite chromosomes and are retained in the new generation. The initial implementations of Multi-objective GA, reported in the literature are of the non-elitist category [116]. For example, David Schaffer suggested one such algorithm namely, Vector Evaluated Genetic Algorithm (VEGA) [120]. The other examples of the non-elitist Multi-objective GA are weight-based Genetic Algorithm [121], Multiple Objective Genetic Algorithm (called MOGA) [122], non-dominated sorting Genetic Algorithm [123], and niched-Pareto Genetic Algorithm [124]. The more recent implementations of the Multi-objective GA are the elitist Multi-objective GA. In general, the elitist Multi-objective GAs are more efficient, since the elitism helps to preserve the best solutions in the past generation and speedup the convergence of the algorithm. Among the elitist Multi-objective GA implementations, some have wide acceptance due to their efficiency in producing better Pareto fronts and the examples are: distance-based Pareto Genetic Algorithm [125], Strength-Pareto Evolutionary Algorithm (SPEA) [126], and Pareto-archived Evolution Strategy [127] and Non-dominated Sorting Genetic Algorithm-II (referred to as NSGA-II in the rest of the thesis). Deb and colleagues [116,128] showed that NSGA-II outperforms the other three algorithms described, in terms of finding a diverse set of solutions and in converging nearer to the true Pareto-optimal set with more efficiency.

The features of Non-dominated Sorting Genetic Algorithm-II (NSGA-II), mentioned above, are useful for the nuclear fuel management related optimization studies. Hedayat and colleagues demonstrated the suitability and efficiency of the NSGA-II in solving the optimization problems of nuclear fuel management [107]. The flavour of the Multi-objective GA considered in the present work is NSGA-II. In the following section, the details of the implementation of the NSGA-II algorithm are considered.

2.7.4. Non-Dominated Sorting Genetic Algorithm-II (NSGA-II)

The procedure followed for the NSGA-II algorithm is shown in the flowchart (Figure 2.14). The procedure of the standard GA like crossover and mutation are the same in NSGA-II also. When NSGA-II is applied to nuclear fuel management application, the fitness evaluation is carried out as usual, by calling the neutronics simulation codes from the algorithm. However, the additional steps for incorporating concepts of Pareto-optimality and dominance are added in the procedure.

The first step of the NSGA-II procedure is to generate the initial parent population, P_t of the size *N*. Then, the fitness evolution is carried out for each member of P_t. During the next step, the crossover and mutation operations are performed on P_t to get offspring population, Q_t. This step is carried out before the fitness evaluation operation of Q_t, and results in getting a new population of size *N*. The combined population of P_t and Q_t (denoted as R_t) undergoes the non-dominated sorting, in the subsequent step of the algorithm. Finally, *N* elements are selected from the combined population and the new population is formed. The whole procedure mentioned above is repeated until the algorithm is getting converged. The non-dominated sorting is used to classify R_t into different Pareto-optimal fronts.

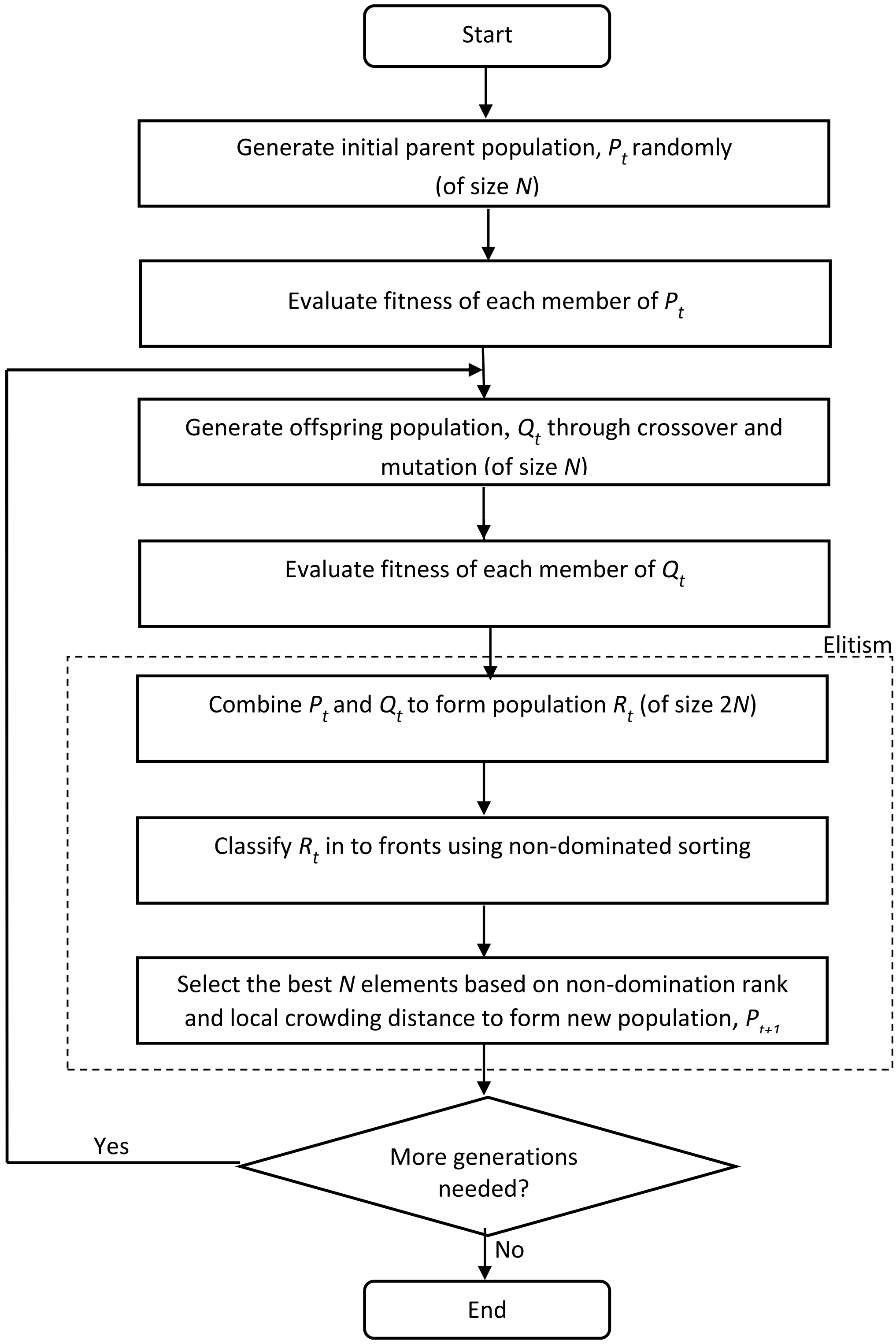


Figure 2.12: Flowchart of Multi-objective GA (NSGA-II implementation). The procedure consists of the standard GA operations and the additional steps of non-dominated sorting and non-domination ranking [116, 119, 128].

According to the concept of dominance, a solution $x^{(1)}$ is said to dominate another solution $x^{(2)}$, if both of the following two conditions are satisfied [119]:

(i) The solution $x^{(1)}$ is not worse than $x^{(2)}$, when all the objectives are considered

(ii) The solution $x^{(1)}$ is strictly better than $x^{(2)}$, in at least one objective

In the first condition, the term "not worse than" indicates that two solutions can equally be good with respect to an objective. The term "strictly better than" in the second condition emphasizes that the equally good solutions are not considered in that case. The solutions belonging to the best Pareto-optimal front, F_1, are the best solutions in the combined population. If the size of F_1, is smaller than *N*, all the members of F_1 are added to the new population, P_{t+1}. The remaining members of P_{t+1}, are chosen from subsequent Pareto-optimal fronts in the order of their ranking. To choose exactly *N* population members, solutions of the last allowed front are sorted using the crowded comparison operator (normally denoted by $<_c$). The new population P_{t+1}, is taken as the input for generating offspring population Q_{t+1}, by applying crowded comparison operator, crossover, and mutation.

The crowded comparison operator assumes that every solution 'i' has two attributes: a non-domination rank, r_i (corresponding to the Pareto-optimal front to which the solution belongs), and a local crowding distance, d_i (a measure of density of solutions in the neighborhood of the Pareto-optimal front). According to the definition of crowded comparison operator, a solution 'i' wins over another solution 'j', if any of the following conditions are satisfied:

(i) If solution 'i' has a better rank than solution 'j', i.e. $r_i < r_j$

(ii) If they have the same rank but solution 'i' has a better crowding distance than solution 'j', i.e. $r_i = r_j$ and $d_i > d_j$

The crowded comparison operator guides the selection process at various stages of the algorithm towards a uniform spread of solutions along the best-known Pareto front. The main advantage of using the crowded comparison operator is that a measure of population density around a solution is computed without requiring a user-defined niche size or the k^{th} closest neighbor [129, 107]. The sorting of the population based on non-domination ranks along with the crowded comparison operation as a diversity-preserving mechanism and provides NSGA-II a powerful 'elitism' strategy. The NSGA-II implementation described above allows the optimization procedure in performing the search process efficiently under the given multiple objectives and constraints.

The nuclear fuel management problems addressed in the present work are implemented by using both the methodologies mentioned above i.e., Penalty-function GA and Multi-objective GA (NSGA-II). This facilitates a comparative study between them, in order to find out the methodology that is more suitable for the particular optimization problem of nuclear fuel management.

2.8. SUMMARY

The overall procedure of the standard GA and the study conducted on the performance of standard GA are covered in the chapter. The different chromosome encoding schemes and operators of the GA, employed in the work carried out as the part of the thesis, are explained. By implementing the GA for solving the Ackley's function based optimization problem, several investigations are conducted in order to understand the influence of various representation schemes and operators on the algorithms performance.

The GA methodologies followed in the work carried out as part of the thesis are discussed. The overall procedure, typically followed when genetic algorithms are applied in the nuclear fuel management, is formulated. The two methodologies in formulating nuclear fuel optimization models for GA, namely Penalty-function GA and Multi-objective GA are discussed. The flavor of Multi-objective GA followed in the present work is Non-Dominated Sorting Genetic Algorithm-II (NSGA-II). Therefore, a detailed description about the procedure and the implementation of NSGA-II is presented in the chapter.

CHAPTER 3

APPLICATION OF GENETIC ALGORITHM IN STEAM CONDENSER OPTIMIZATION OF PROTOTYPE FAST BREEDER REACTOR

This chapter presents an engineering design optimization study, conducted on the steam condenser of a fast breeder reactor. In the present study, the standard GA (real-parameter) is applied for solving a single objective problem with limited constraints. As part of the optimization procedure, the GA based performance-cost analysis of the circulating water system is carried out. The purpose of the study is to demonstrate the suitability of the standard real-parameter GA in the engineering design application of a reactor subsystem.

3.1. INTRODUCTION

The 500 MWe Prototype Fast Breeder Reactor (PFBR) is designed by Indira Gandhi Centre for Atomic Research (IGCAR) and is being commissioned at Kalpakkam, India [130]. Prototype Fast breeder reactor (PFBR) is a sodium cooled, pool type, mixed-oxide fuelled reactor with two secondary loops [131]. The primary objective of PFBR is to demonstrate the techno-economic viability of fast breeder reactors on an industrial scale. The design optimization of the steam condenser of the

PFBR is considered in the present work. The two important functions of steam condenser are:

(i) condense the exhaust steam from the steam turbine to get better efficiency

(ii) convert the turbine exhaust steam into pure water so that it may be reused in the steam generator as boiler feed water

Therefore, the steam condenser is an important part of the Circulating Water System of a nuclear power plant. The Circulating Water System of PFBR is once-through cooling type system, where water from the sea is pumped to the steam condenser and is returned back to the sea.

The proper design of the Circulating Water System (referred to as CWS in the rest of the paper) in a power plant improves the electrical output for a given input heat energy. Nearly 2/3rd of the heat energy generated in the power cycle is ultimately rejected to the atmosphere in the form of waste heat through the CWS. Lowering of the saturation temperature of the exhaust steam through the efficient design of the steam condenser can improve the efficiency of the total CWS. The design parameters of the steam condenser which are influencing the efficiency of the CWS are selected for the study. The optimization procedure is based on the performance-cost analysis carried out by applying the optimization concepts of GA. The optimal values for the design parameters are arrived from the study, in order to get the maximum capitalized profit from the CWS.

The Circulating Water System of Prototype Fast Breeder Reactor (PFBR) is designed based on the conventional design optimization concepts [132]. The design parameters are arrived based on the calculation approach in which the parameter values are incremented or decremented in every iteration of the optimization procedure. A study has been conducted based on the above approach by Sen [132]. In the present work, the parameters of the optimization problem and their specifications are adopted from that reference study. The results obtained from the present work are compared with the results reported from the reference study.

The optimization problem - the design of Circulating Water System (CWS) - considered in the present study is of less computational complexity in terms of objective search space and parameter evaluation procedure. The aim of the study is to extend the application of the standard GA procedure and the GA operators applied for the benchmark problem (Ackley's function optimization presented in Chapter 2) to a relevant engineering application of a reactor subsystem. In the study, the suitability of the standard real-parameter GA in the engineering design application of a reactor subsystem is demonstrated.

3.2. PROBLEM DESCRIPTION AND OPTIMIZATION MODEL FORMULATION

The Circulating Water System (CWS) of Prototype Fast Breeder Reactor (PFBR) consists of steam surface condenser (single pass shell and tube heat exchanger), circulating water pumps and drives, pump house, piping/ducts, seal well, outfall structure etc. Direct cooling system with seawater is envisaged for the PFBR. Usually for a direct cooling system, a single pass condenser is a better configuration, from techno-economic point of view, than a two-pass condenser [133]. Therefore, a single pass condenser has been considered in this study. The schematic of CWS of PFBR is shown in Figure 3.1. The seawater is drawn through a submarine tunnel to the onshore pump house. The pump house has screening mechanisms for filtering seawater before it enters the pumps. It houses two numbers of condenser cooling water pumps and two numbers of auxiliary seawater pumps. Water is pumped to the steam condenser by the condenser cooling water pumps and to the process heat exchangers by the auxiliary seawater pumps.

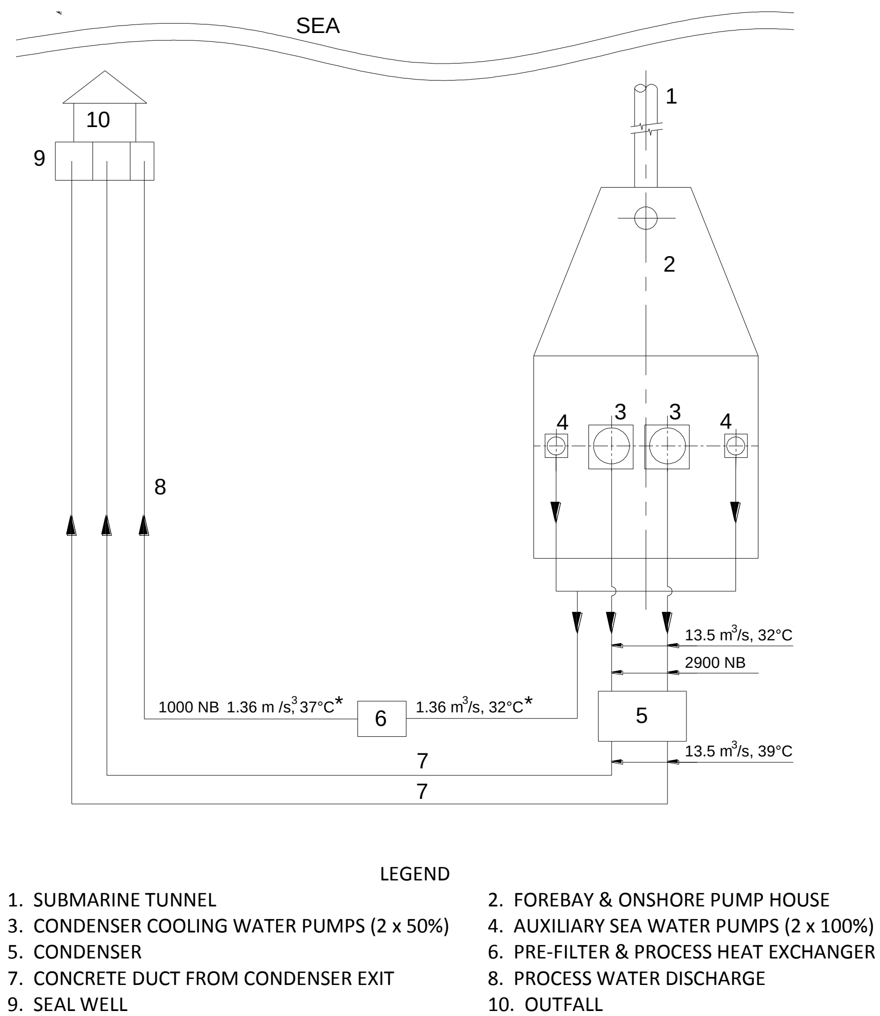


Figure 3.1: Schematic of circulating water system of Prototype Fast Breeder Reactor (PFBR).

The discharge of the condenser is led to the seal pit through subsurface concrete ducts, while the discharge of the process heat exchangers is let to the seal pit through the buried discharge pipes. The water is discharged to the sea from the outfall.

During the optimization procedure, four design parameters of the condenser are considered and the profit generated by the total Circulation Water System (CWS) for the given values of design parameters are calculated. The parameters which are selected as design candidates for this study are:

(i) condenser flow rate (Q)

(ii) outer diameter of condenser tube (*do*)

(iii) condenser tube length (*l*)

(iv) velocity of water inside the tube (*v*)

The aim of the optimization procedure is to find the optimum values for the above parameters that would provide highest capitalized profit. The environmental regulation stipulates that the hot water released to the sea should not exceed 7 degree above the ambient temperature of the intake seawater. Therefore, the feed-water temperature rise or the temperature range (denoted as '*r*') of the CWS is fixed accordingly. Another constraint in the study is related to the terminal temperature difference (denoted as '*TTD*') of the condenser which is defined as the saturation temperature of the extraction steam minus the feed-water outlet temperature. An increase in *TTD* indicates a reduction in the heat transfer, while a decrease indicates an improvement. The recommended lower limit of *TTD* for the condenser is 2.78 degree [134]. Therefore, if the rise of water temperature in the condenser exceeds 7 degree or the terminal temperature difference for the condenser is less than 2.78 degree, then that design candidate is rejected during the optimization.

Considering the given design parameters and the associated constraints, a mathematical model formulation of the given optimization problem is arrived at as follows:

$$Max\ (\text{Capitalized profit}) = f\,(Q, d_o, l, v) \qquad (3.1)$$

Such that,

$$r < 7 \text{ degree}, \quad TTD > 2.78 \text{ degree} \qquad (3.2)$$

In the above representation *f* () represents the fitness function with the parameters involved as the arguments and *Max* represents the maximization. The problem has the following boundary conditions for the design parameters:

$$96000.00 \leq Q \leq 105000.00 \tag{3.3}$$

$$d_o = 22.225 \text{ or } 25.400 \tag{3.4}$$

$$12 \leq l \leq 18 \tag{3.5}$$

$$2.10 \leq v \leq 2.70 \tag{3.6}$$

The boundary conditions given above are arrived by considering the results of the reference study [132]. During the optimization procedure, the condenser flow rate (Q) is assigned with real number values from the given range (the assigned precision is of two decimal places). The outer diameter of condenser tube (d_o) can take either of the given two values during the optimization procedure i.e. either 22.225 or 25.400. This constraint is based on the commercially available condenser tube specifications. The condenser tube length (l) is assigned with integer values in the given range. The velocity of water inside the tube (v) can take any real number value with the assigned precision of two decimal places, from the specified range. The values of constraints given in the Eqn. 3.2 (i.e. values of 'r' and 'TTD') are calculated during the fitness evaluation of the optimization procedure. A detailed description about the procedure of fitness evaluation is given in the following section.

3.3. THE FITNESS EVALUATION PROCEDURE

The given optimization problem is formulated as a profit maximization problem as described above. During the optimization procedure, the fitness evaluation function is called by the GA, for finding the capitalized profit values. The fitness evaluation is carried out for every individual in the population of GA, with the four decision parameters as the chromosome elements. The evaluated capitalized profit values (or the fitness values) are assigned to the corresponding individual. The procedure used for fitness evaluation is divided in to two, namely, calculation of performance and calculation of profit.

3.3.1. Calculation of the System Performance

The methodology used for calculation of the performance of the Circulating Water System (CWS) is described as follows. For the selected chromosome, the performance of the system is evaluated for a given cooling water temperature based on the thermodynamic relationship between the condenser backpressure and turbine output. The calculation takes into account the correction factors for the tube diameter, inlet seawater temperature, tube material and gauge factor, and tube cleanliness. The term 'performance' means calculation of the saturation temperature and corresponding pressure of the condensing steam at the turbine exhaust and hence the power generated and corresponding revenue earned.

During the above performance calculation, if the solution violates constraints regarding the rise of condenser inlet-outlet temperature, '*r*' and the difference of terminal temperature, '*TTD*', then the performance calculation is repeated by rejecting the present solution and replacing the gene values with randomly selected new set of values. The above step is continued until the viable solution candidates are obtained from performance calculation. Then, the next steps are carried out for calculating the capitalized profit, associated with every solution set. The basic equations involved in the performance calculation are summarized in Table 3.1. The flowchart representing the computational procedure followed in the performance calculation is given in Figure 3.2.

3.3.2. Calculation of the Capitalized Profit

The term 'capitalized profit' indicates the profit added by the Circulating Water System (CWS) over a period of years to the capital. The calculation of capitalized profit involves two components, i.e. calculation of the system cost and calculation of the capitalized revenue. The cost of CWS for each design candidate is calculated first.

Table 3.1: Basic equations involved in the performance calculation of the Circulating Water System (CWS)

$$H = P.1000.(H_r - 860) \quad (3.7)$$
$$H = \gamma.Q.r.C_p \quad (3.8)$$
$$Q = 3600.(\pi/4).d_i^2.v.\, n_t \quad (3.9)$$
$$A = \pi\, .\, d_o l.\, n_t.\, n_p \quad (3.10)$$
$$H = U.\, A.\, (LMTD) \quad (3.11)$$
$$U = U_1.\, F_w.F_m.\, F_c \quad (3.12)$$
$$LMTD = \frac{HWT - CWT}{ln[(Ts - CWT)/(Ts - HWT)]} \quad (3.13)$$
$$TTD = r/[exp\{r.\, U.\, A/H\}-1] \quad (3.14)$$
$$r = HWT\text{-}CWT \quad (3.15)$$
$$T_s = HWD+TTD \quad (3.16)$$

Parameter	Meaning (unit)	Parameter	Meaning (unit)
H	Heat rejection rate (kcal/hr.)	A	Surface area of the Condenser (m^2)
P	Electrical power output (MW)	$LMTD$	Log mean temperature difference between water and steam (K)
H_r	Heat rate of power cycle (kcal/kWhr.)	U_1	Uncorrected heat transfer coefficients
γ	Specific weight of sea water (kg/m^3)	F_w	Inlet water temperature correction factor
Q	Circulating water flow-rate (m^3/hr)	F_m	Tube material and gauge correction factor
l	Length of tube (m)	F_c	Cleanliness correction factor
C_p	Specific heat of sea water	v	Velocity of water inside the tube (m/sec)
d_i	Condenser tube inner diameter (m)	HWT	Hot water temp. at condenser outlet (K)
d_o	Condenser tube outer diameter (m)	CWT	Cold water temp. at condenser inlet (K)
n_t	Number of tubes per pass	T_s	Saturation temperature of steam (K)
n_p	Number of passes in the condenser	r	Rise in feed water temperature (K)
U	Overall heat transfer coefficient ($kcal/m^2hrK$)	TTD	Terminal temp. difference (K)

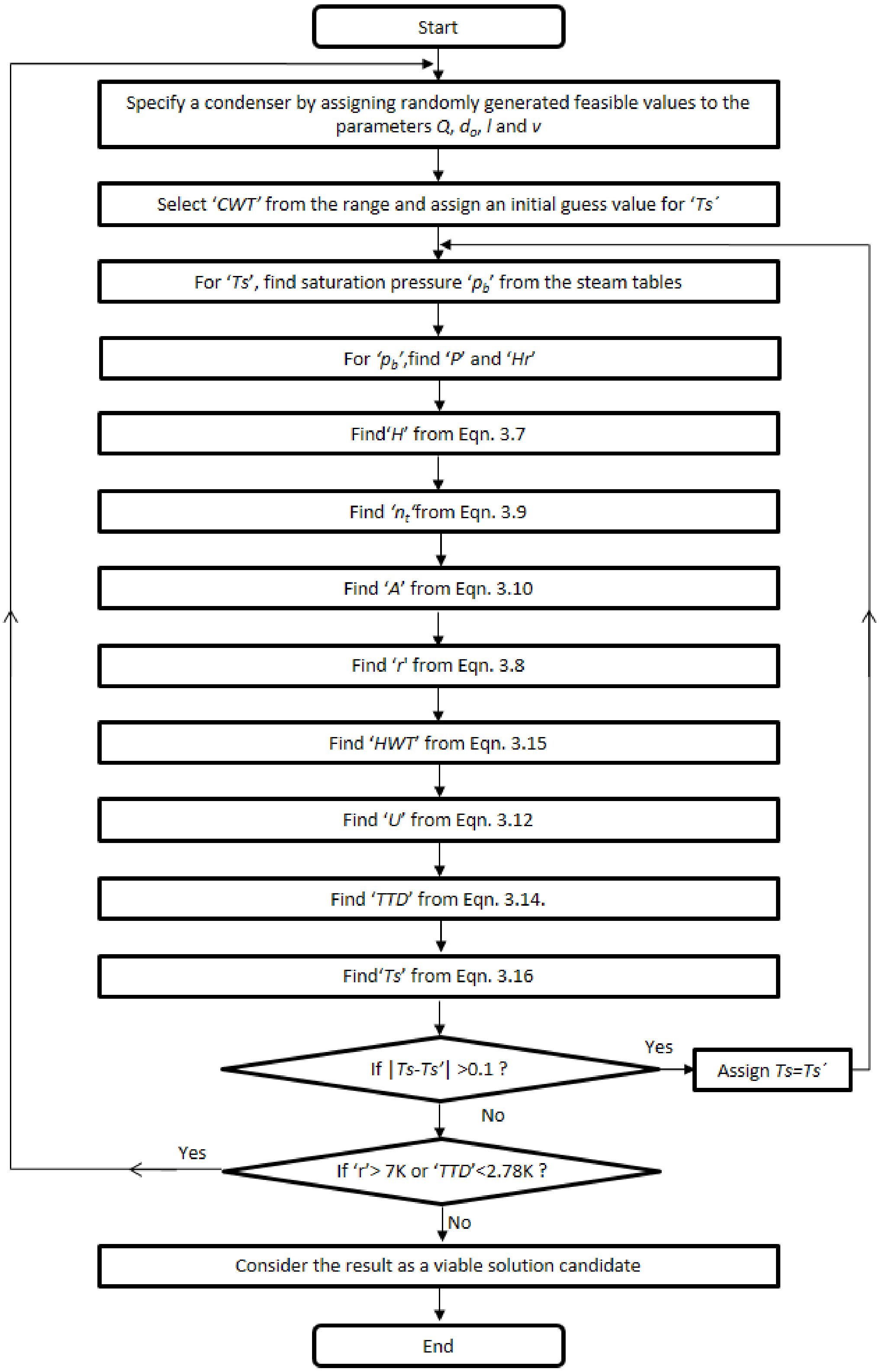


Figure 3.2: The computational procedure followed in the performance calculation of the Circulating Water System (the equations given in Table 3.1 are referred in the flowchart)

The investment cost of equipment and the capitalized operating cost of circulating water pumps are considered for the cost calculation. Then the capitalized revenue earned is calculated based on the electricity generated from the plant. Finally, the capitalized profit is calculated as capitalized revenue minus cost of CWS. All costs and economic figures in the present work are in lakhs of Indian rupees (referred to as Rs in the rest of the study). The lifetime of the system is considered as 40 years, the amortization rate is 14.5%, energy cost of Rs 3.25/- unit at the rate of increase of 3% per year and pump efficiency of 87%. The detailed description of steps involved in the calculation of capitalized profit is given below.

3.3.2.1. Cost of the Circulating Water System

The components of costs which determine the cost of the Circulating Water System (CWS) are:

(i) investment cost

(ii) operational cost

These costs are calculated as follows:

(i) Investment Cost

The investment cost (denoted as $'C_{IN}'$) has three major components and are described below:

- **Cost of the surface condenser**

$$\text{Cost of the condenser} = A \times c_A \tag{3.17}$$

where, A = Surface area of the condenser (m^2)

c_A = Cost per unit surface area (Rs /m^2)

- **Cost of the Circulating Water System pumps and drives**

$$\text{Cost of the pumps and drives} = N_p \left[\frac{9.81\, Q_p\, \gamma\, H_p}{1000\, \eta_P\, \eta_M}\right] CP_{kw} \qquad (3.18)$$

where, N_p = Number of CWS pumps

Q_p = Flow rate through each pump (m^3/sec)

γ = Specific weight of seawater (kg/m^3)

H_p = Pumping Head (m)

η_P = Efficiency of pump

η_M = Efficiency of motor

CP_{kw} = Cost of the pump drive per kW (Rs/kW)

- **Cost of the pump house, tunnel, forebay and ducts/pipes**

The costs of pump house, tunnel, forebay and CWS duct/pipe are not fixed, i.e. costs will vary according to the flow rate of water through the CWS. The costs of the components mentioned above, for different flow rates given, have been taken from the reference study [132]. The cost of in-between flow rates are interpolated as part of the GA based optimization procedure. The cost of seal well and outfall structure has not been considered in the study.

(ii) Operational Cost

The major component of the operational cost is attributed to the energy consumption by the pumps of CWS, which is calculated as:

$$C_{OP} = C_f\, E_e\, P_s \qquad (3.19)$$

where,

C_{OP} = Operational cost (based on the capitalized cost of energy consumption by the pumps)

C_f = Capitalization factor (function of plant life, interest rate and rate of rise of cost of electricity)

E_e = Yearly energy consumption (kW hr) by the pumps

= $e_p \, N_H$

P_s = Unit selling price of electricity (Rs / kWhr)

N_H = Number of hours of plant operation per year

e_p = Electrical power required by the pumps (kW)

= $N_r \left[\frac{9.81 \, Q_p \, \gamma \, H_p}{1000 \, \eta_P \, \eta_M}\right]$

N_r = Number of pumps running

3.3.2.2. Calculation of the Capitalized Revenue Earned

After the calculation of the cost of the Circulating Water System, the next step is the calculation of capitalized revenue earned from the system, based on the total power produced in a year. It is assumed that 'P_1' is the power produced (in MW) for a particular water temperature at the condenser inlet (i.e. the cold water temperature 'CWT') which occurs for $'n_1'$ hours. Then, the electricity generated during the period is given by:

$$e_1 \text{ (in kWh)} = 1000 \, P_1 \, n_1 \tag{3.20}$$

The total electricity generation is calculated based on the number of hours of plant operation and the total number of hours of plant operation (denoted as 'N_H') is calculated as:

$$N_H = n_1 + n_2 + n_3 + \cdots + n_n \tag{3.21}$$

where, $n_1, n_2 \ldots n_n$ represents different time periods of plant operation at different power levels.

Total electricity generation in a year, $'E_T'$ is calculated by summing up the electricity generation for all the time periods, as given below:

$$E_T \quad = e_1 + e_2 + e_3 + \cdots + e_n \qquad (3.22)$$

In the above equations, $'e_1'$ represents the electricity generated for $'n_1'$ hours, $'e_2'$ represents the electricity generated for $'n_2'$ hours, and so on.

Let $'P'_s$ be the unit-selling price of electricity in Rs/kWh. Then,

$$R_C \quad = C_f E_T P_s \qquad (3.23)$$

where,

R_C = Capitalized revenue earned

C_f = Capitalization factor

E_T = Total electricity generation in a year

P_s = Unit selling price of electricity (Rs / kWhr)

3.3.2.3. Calculation of the Total Cost

The investment cost and operational costs are added to get the total cost (denoted as $'C_{TOT}'$) of the system as follows:

$$C_{TOT} = C_{IN} + C_{OP} \qquad (3.24)$$

where,

C_{IN} = Investment cost

C_{OP} = Operational cost

3.3.2.4. Calculation of the Capitalized Profit

The capitalized profit is obtained by subtracting the total cost (C_{TOT}) from the capitalized revenue (R_C), and is represented as:

$$\text{Capitalized profit} = R_C - C_{TOT} \qquad (3.25)$$

The capitalized profit calculated is assigned as the fitness value of the individual solution candidate (or chromosome) of the algorithm. The implementation details of the algorithm are covered in the next section.

3.4. DETAILS OF IMPLEMENTATION

The design parameters (Q, l, d_o, v) selected for the study form the decision variables for the GA based optimization procedure. All the four decision variables are real numbers and hence the real-number representation is followed in the chromosome representation of the algorithm. The GA parameters, operators and methods considered for the study are summarized in Table 3.2. The operators and their values given in the table are arrived at, based on the results of the study carried out on the benchmark optimization problem addressed in Chapter 2.

As an example, two chromosomes generated by the algorithm are shown in Figure 3.3. The chromosome representations follow the order of the design parameters i.e., Q, d_o, l and v. The values assigned to the chromosomes in the example are taken from the first generation of a trial run (Trial No. 1). The third chromosome element that represents the parameter '*l*' is defined as a real number variable in the implementation, but always gets truncated to the corresponding integer value, before storing to the chromosome representation.

Table 3.2: Genetic parameters and methods or values used in the GA based optimization procedure.

Parameter	Methods/Values
Chromosome representation	Real-number
Population size	50
Crossover Method	Arithmetical
Crossover Probability (CR)	0.8
Mutation Method	Non-uniform
Mutation Probability (MR)	0.01
Maximum number of Generations	100
Number of Elite members (*EL*)	5

chromosome 1

103166.75	**25.400**	**17.00**	**2.58**
Q	d_o	l	v
98594.01	**22.225**	**12.00**	**2.12**

chromosome 2

Figure 3.3: Examples of two chromosomes generated by the algorithm. The same chromosomes are considered in the illustration of the crossover operation.

As the crossover method suitable for real-number representation is arithmetical crossover, the same method is employed in the present work. The arithmetical crossover method is applied for the first, third and the fourth elements of the chromosome which represents the parameters Q, l and v respectively. However, for the second element of the chromosome, that represents the parameter d_o, the crossover operation is achieved by the 'swapping' of elements.

The crossover based on 'swapping', satisfies the special requirement of the parameter 'd_o', i.e., it can take only two values, either 25.400 or 22.225. This constraint is based on the commercially available specifications of condenser tubes. Once two parents are selected, all the four elements of parents are undergoing the crossover operation. In order to illustrate operations of the algorithm, the chromosomes given in the example (Figure 3.3) can be represented as two chromosome vectors as follows:

$$x_1 = (103166.75,\ \ 25.4,\ \ 17.0,\ \ 2.58\ \ ) \qquad (3.26)$$

$$x_2 = (98594.01,\ \ 22.225,\ \ 12.0,\ \ 2.12) \qquad (3.27)$$

Let us now assume that the above chromosome vectors are selected as parents for the crossover operation. The arithmetical crossover operation performed on the first elements of the parent chromosome (i.e. on 103166.75 and 98594.01) can be represented as:

$$t_{11} = (a \times 103166.75 \ + (1 - a) \times 98594.01) \qquad (3.28)$$

$$t_{12} = (a \times 98594.01 \ \ + (1 - a) \times 103166.75) \qquad (3.29)$$

(by following the equations which define the arithmetical crossover, given in Chapter 2, i.e. Eqns. 2.2 and 2.3)

where, t_{11} and t_{12} represent the first two elements in the resulting offspring and 'a' is the random number generated between 0 and 1. As mentioned earlier, the crossover of the second element of the parent chromosomes is achieved by just swapping the contents. The arithmetical crossover happening on the third elements of the parent chromosome (i.e. on 17.0 and 12.0) can be represented as:

$$t_{31} = (a \times 17.0 \ + (1 - a) \times 12.0) \qquad (3.30)$$

$$t_{32} = (a \times 12.0 \ + (1 - a) \times 17.0) \qquad (3.31)$$

where, t_{31} and t_{32} represents the third two elements in the resulting offspring and 'a' is the random number generated between 0 and 1. Similarly, the arithmetical crossover performed on the fourth element of the parent chromosome (i.e. on 2.58 and 2.12) can be represented as:

$$t_{41} = (a \times 2.58 \ + (1 - a) \times 2.12) \qquad (3.32)$$

$$t_{42} = (a \times 2.12 \ + (1 - a) \times 2.58) \qquad (3.33)$$

where, t_{41} and t_{42} represents the fourth two elements in the resulting offspring and 'a' is the random number generated between 0 and 1. At a particular instance, for 'a' = 0.6358, then the resultant offspring chromosomes, 'o_1' and 'o_2' (arrived based on Eqns. 3.28 to 3.33) are:

$$o_1 = (101501.36, \ 22.225, \ 15.0, \ 2.41) \qquad (3.34)$$

$$o_2 = (100259.40, \ 25.4, \ 13.0, \ 2.29) \qquad (3.35)$$

The arithmetical crossover operation explained above is illustrated in the Figure 3.4. The examples of two chromosomes generated by the algorithm are considered as the parents. The crossover operation generates the children as shown in the figure.

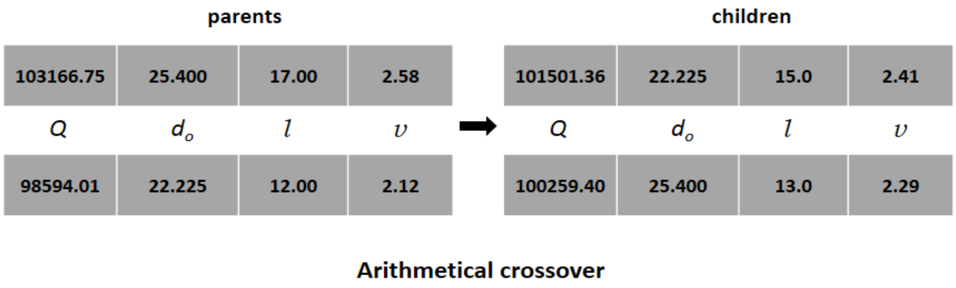


Figure 3.4: Illustration of the arithmetical crossover operation. The two chromosomes generated by the algorithm are considered as parents. The crossover operation generates the children as shown.

The next operation that comes under the procedure of GA is the mutation. The mutation method followed in the implementation is the non-uniform mutation. In order to avoid getting infeasible values, the mutation operation is not performed on the parameter 'd_o'. In continuation of the example considered above, assuming that one of the offspring (o_1) derived from crossover operation is undergoing the non-uniform mutation at 60th generation (t=60). It is assumed that, the first element of 'o_1' (that represents the parameter 'Q') is chosen for the mutation operation (i.e. $v_k = 101501.36$). The given upper and lower bounds for the parameter 'Q' are 105000.00 and 96000.00 respectively. If the random digit generated, as part of the non-uniform mutation operator is 0, then the mutated element v_k^l is calculated as:

$$v_k^l = 101501.36 + \Delta\,[60, (105000.00 - 101501.36)] \qquad (3.36)$$

(by following the equations which define the non-uniform mutation, given in Chapter 2, i.e. Eqn. 2.4 and 2.5)

Now, if $r = 0.80$, then $\Delta\,(t, y)$ function in the above step will return 2239.13 and v_k^l will get 103740.49. In essence, for the considered example, the non-uniform mutation operation changes the first chromosome element's value from 101501.36 to 103740.49. The non-uniform mutation operation explained above is illustrated in the Figure 3.5.

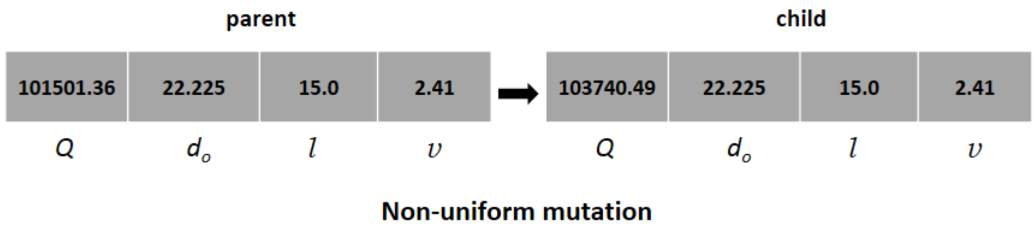


Figure 3.5: Illustration of the non-uniform mutation operation. In the example, the first chromosome generated by the crossover operation, is considered as the parent. The value representing the condenser flow rate (Q) is modified during the mutation operation.

The ‘C’ programming language is used for the implementation of algorithm. The initial population is generated by calling the random number generation function of the C programming language. Using the function, random numbers are generated within the range of boundary conditions for the specific values of decision parameter. The selection method followed is Roulette-Wheel method (see Section 2.4.2. for details). The concept of *Elitism* is also incorporated in the selection method of GA, where a fixed number of the chromosomes having higher fitness values are considered as elite chromosomes and are retained in the new generation. The results generated by the optimization procedure are discussed in the next section.

3.5. RESULTS AND DISCUSSIONS

As part of the tuning of algorithm parameters, several trial runs were conducted with randomly generated initial population of the GA. Each trial run was started with entirely different initial population ensuring different initial search space for different trial runs. Based on the results generated initially, the GA parameters were fine-tuned and the final parameters used in the study are arrived at. The fine-tuned version of the algorithm is used to generate final results by conducting eight independent trial runs. The best result (i.e. with maximum capitalized profit) of the final generation population is extracted for every trial run and is presented in Table 3.3. The economic figures (i.e., total cost, capitalized revenue and capitalized profit) arrived at from the optimization procedures are also presented in the table. These economic figures are used in analyzing the performance of the algorithm performance.

The results reported by the reference study [132] and the results arrived from the present optimization study are given in Table 3.4. The summary of results presented in the table indicate the values or range of values to be assigned to the design parameters of the condenser, for getting the maximum capitalized profit from the Circulating Water System (CWS). The following observations can be arrived from the results of the trial runs (Table 3.3) and from the summary of results (Table 3.4):

- The GA based optimization procedure is able to generate feasible design parameter values which fall in line with the results of the reference study. All the four design parameter values (i.e. Q, l, d_o, v) generated by the algorithm fall in the range of results of the reference study.
- Different trial runs generate different design parameter values (with in the range of optimal solutions) and corresponding economic parameter values. This behavior is due to the stochastic nature of the GA.

3.5.1. Performance of the Algorithm

One of the common methods of assessing the performance of GA is by measuring the average fitness value in each generation [113]. The capitalized profit represents the fitness value for the algorithm. The average fitness is calculated by adding the individual fitness values calculated by the fitness evaluation step of the algorithm, and dividing it by the population size. Since the present study is with the objective of profit maximization, achieving the higher values of average fitness is desired.

Table 3.3: The results obtained by the present study for the design parameters and the corresponding economic figures. The best results from the final population of every trial run are presented

Trial No.	Q	d_o	l	v	Total Cost (in lakhs of Rs)	Capitalized revenue(in lakhs of Rs)	Capitalized Profit(in lakhs of Rs)
1	98334.15	25.4	15	2.11	30484.69	816746.81	786262.12
2	98940.37	25.4	15	2.10	31274.31	817082.37	786129.82
3	98657.38	25.4	15	2.10	30846.38	816853.61	786207.03
4	98077.18	25.4	15	2.11	30401.88	816721.23	786319.36
5	98216.48	25.4	14	2.10	29427.47	814811.94	786230.78
6	98022.23	25.4	16	2.10	30831.23	817048.85	786217.63
7	98418.05	25.4	16	2.10	30977.28	817079.22	786101.94
8	98487.48	25.4	15	2.10	30526.72	816765.15	786238.44

Table 3.4: The summary of results obtained from the reference study* and the present study

Parameter	Unit	Results (value/range) Reference study*	Results (value/range) Present Study
Condenser flow rate (Q)	m^3/hr	98,000.00 to 99,000.00	98,022.23 to 98,940.37
Outer diameter of condenser tube (d_o)	mm	25.4	25.4
Condenser tube length (l)	m	14.0 - 16.0	14.0 - 16.0
Velocity of water inside the tube (v)	m/s	2.10	2.10 - 2.11

**Doc No. PFBR/71200/DN/2053 Rev.4(2007)* [132]

The generation wise evolutions of the average fitness for the eight trial runs are shown in Figure 3.6. The overall trend is for increasing the average fitness values towards the end of generations. Even though, all the trial runs are able to improve their average fitness values towards the final generations, the variations in average fitness are randomly high between the generations. The reason for this behavior can be explained as follows. According to the procedure of calculation of the system performance (see Section 3.3.1), a rejection strategy is followed when a solution candidate does not obey the constraints related to the rise in water temperature (r) and the terminal temperature difference (TTD). In that case, a new solution candidate is randomly generated and the performance calculation is started afresh. The fitness value evaluated for the new solution candidate can fluctuate substantially from the average fitness of the previous generation. When the numbers of such new solution candidates are more in a generation, the average fitness of that generation fluctuates randomly with respect to the previous generation.

Next, the generation wise evolution of the best fitness is considered. The best fitness represents the candidate solution in a generation with maximum capitalized profit. The improvement in best fitness is an important property of the algorithm. The generation wise evolutions of the best fitness for the eight trial runs are shown in Figure 3.7. The best fitness values are getting improved towards the end of the generations in all the trial runs. The ‘elitism’ helps in achieving this feature by the algorithm. Once a better solution candidate is generated by the standard GA operators, that solution is preserved and given forward to the subsequent generations by the ‘elitism’.

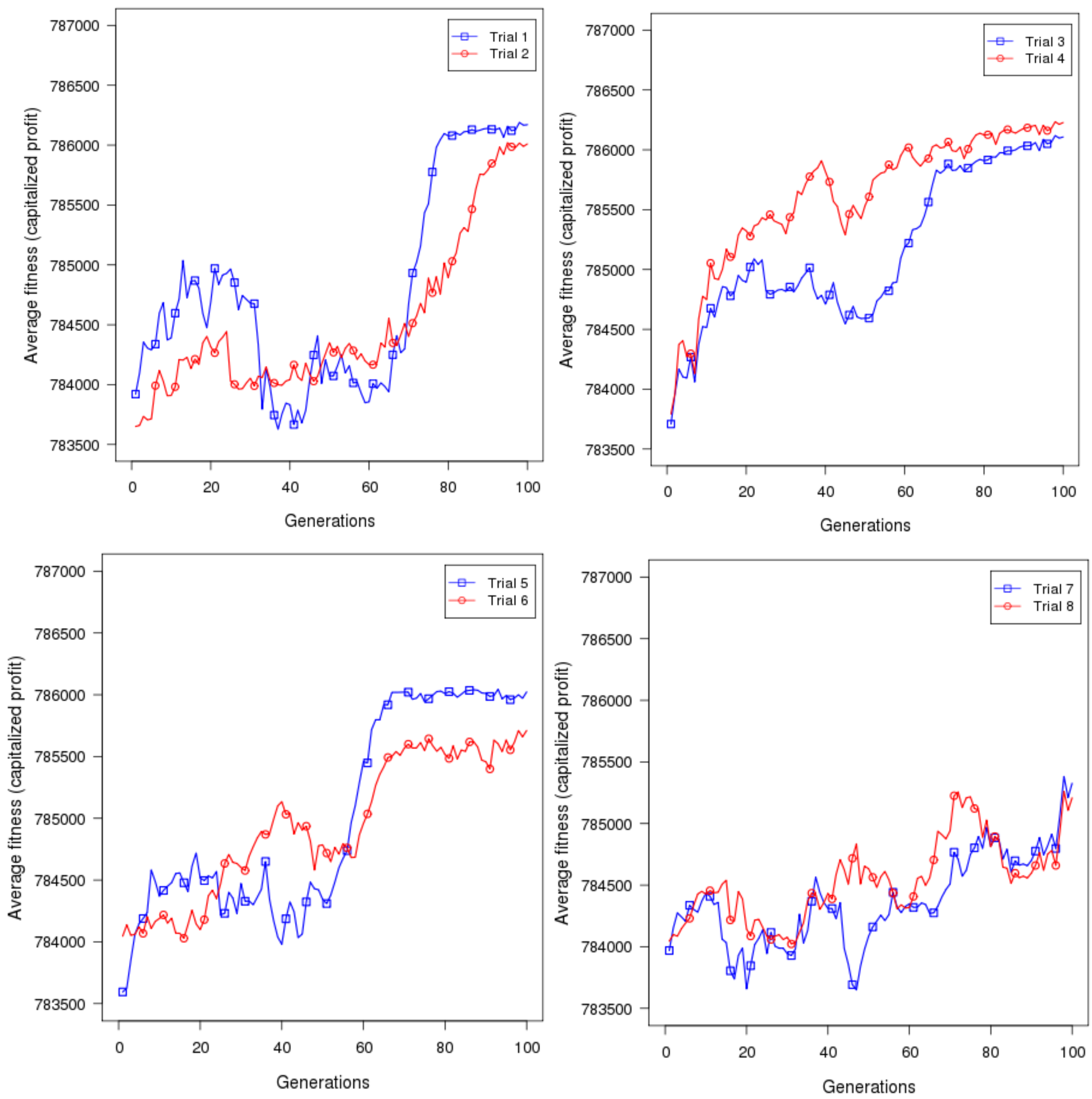


Figure 3.6: The generation wise evolutions of the average fitness (or average values of the capitalized profit in lakhs of Indian rupees) for different independent trial runs.

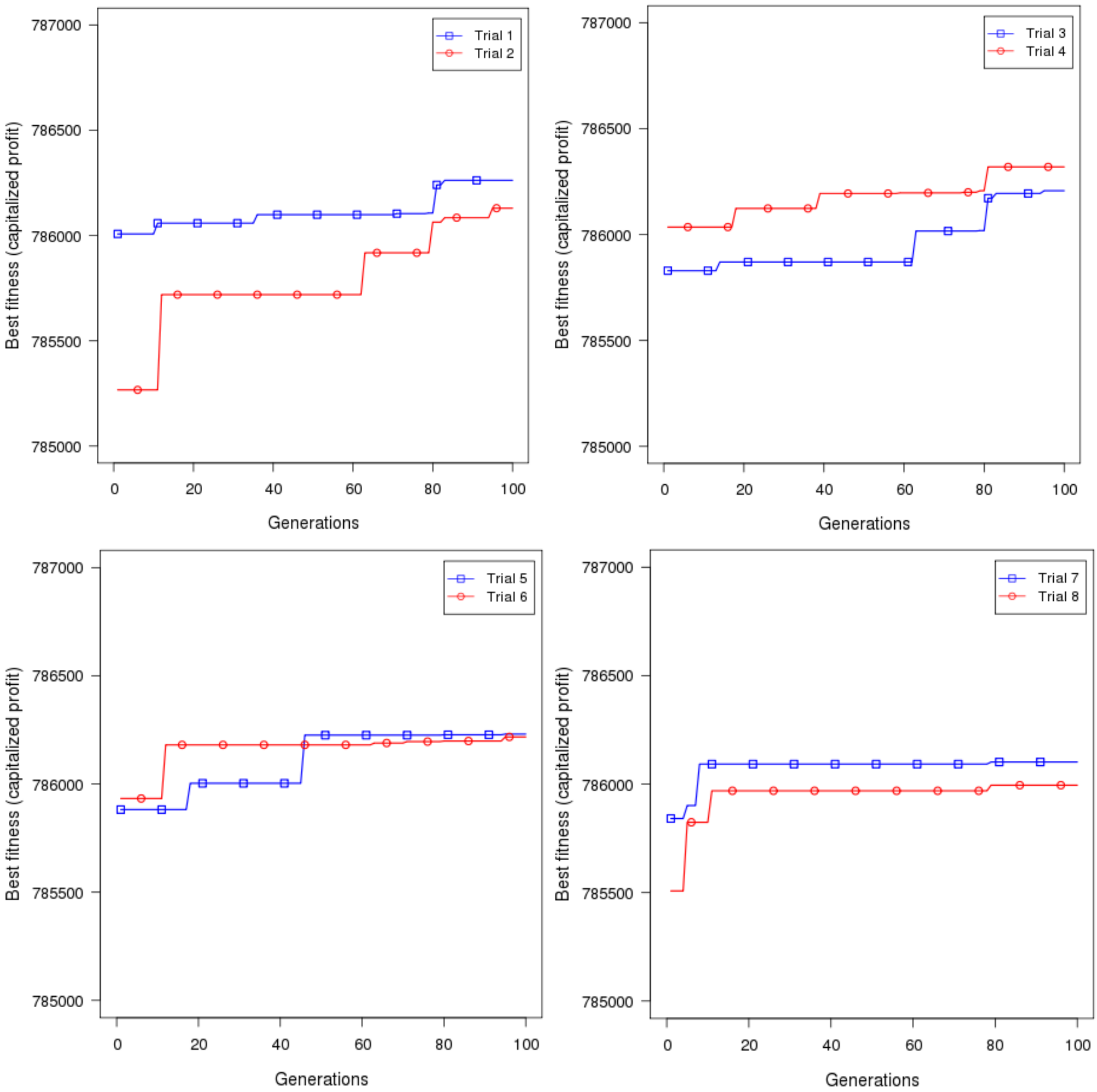


Figure 3.7: The generation wise evolutions of the best fitness (or best values of the capitalized profit in lakhs of Indian rupees) for different independent trial runs.

3.5.2. Influence of Chromosome Elements on Economic Figures

The chromosome elements of the algorithm are constituted by the four design parameters (i.e. Q, l, d_o, v). During the evolution of the algorithm, different sets of values are assigned to the design parameters by the algorithm. The total cost, the capitalized revenue and the capitalized profit are calculated based on the values assigned to the design parameters. The objective of the optimization is to maximize the capitalized profit. In order to achieve this objective, there are different possibilities like, minimizing the total cost or maximizing the capitalized revenue or doing both together. The algorithm will automatically select the one which comes under the influence of its chromosome elements. In order to find out the influence, the evolution of total cost and capitalized revenue for the eight independent trial runs are shown in Figures. 3.8 and 3.9 respectively. It can be observed from Figure 3.8 that the total costs are getting minimized towards the end of generations. However, there is no trend of maximization for the capitalized revenue (Figure 3.9). Therefore, it can be concluded that the chromosome elements have more influence on the total cost than on the capitalized revenue.

3.6. SUMMARY

In the chapter, the application of the standard GA procedure in the optimization of design parameters for the steam condenser of Prototype Fast Breeder Reactor (PFBR), is considered. The real-number encoding, arithmetical crossover, non-uniform mutation and elitism are applied in the GA implementation. The design parameters considered for the optimization procedure are condenser flow rate (Q), outer diameter of condenser tube (d_o), condenser tube length (l) and velocity of water inside the tube (v). The result obtained from the study shows that the design parameter values generated by the algorithm fall in the range of results of the reference study.

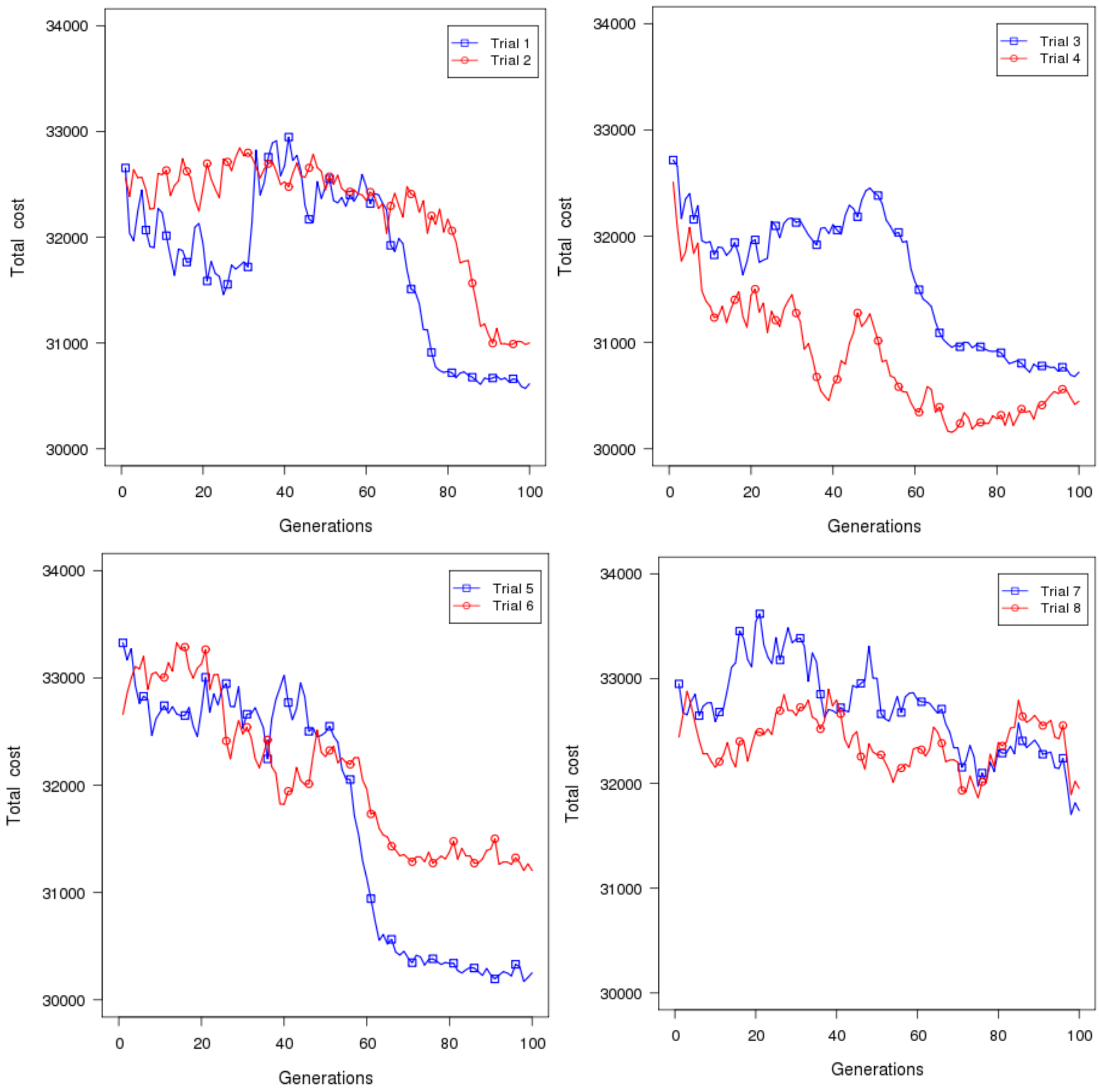


Figure 3.8: The generation wise evolution of total cost (in lakhs of Indian rupees) for different independent trial runs

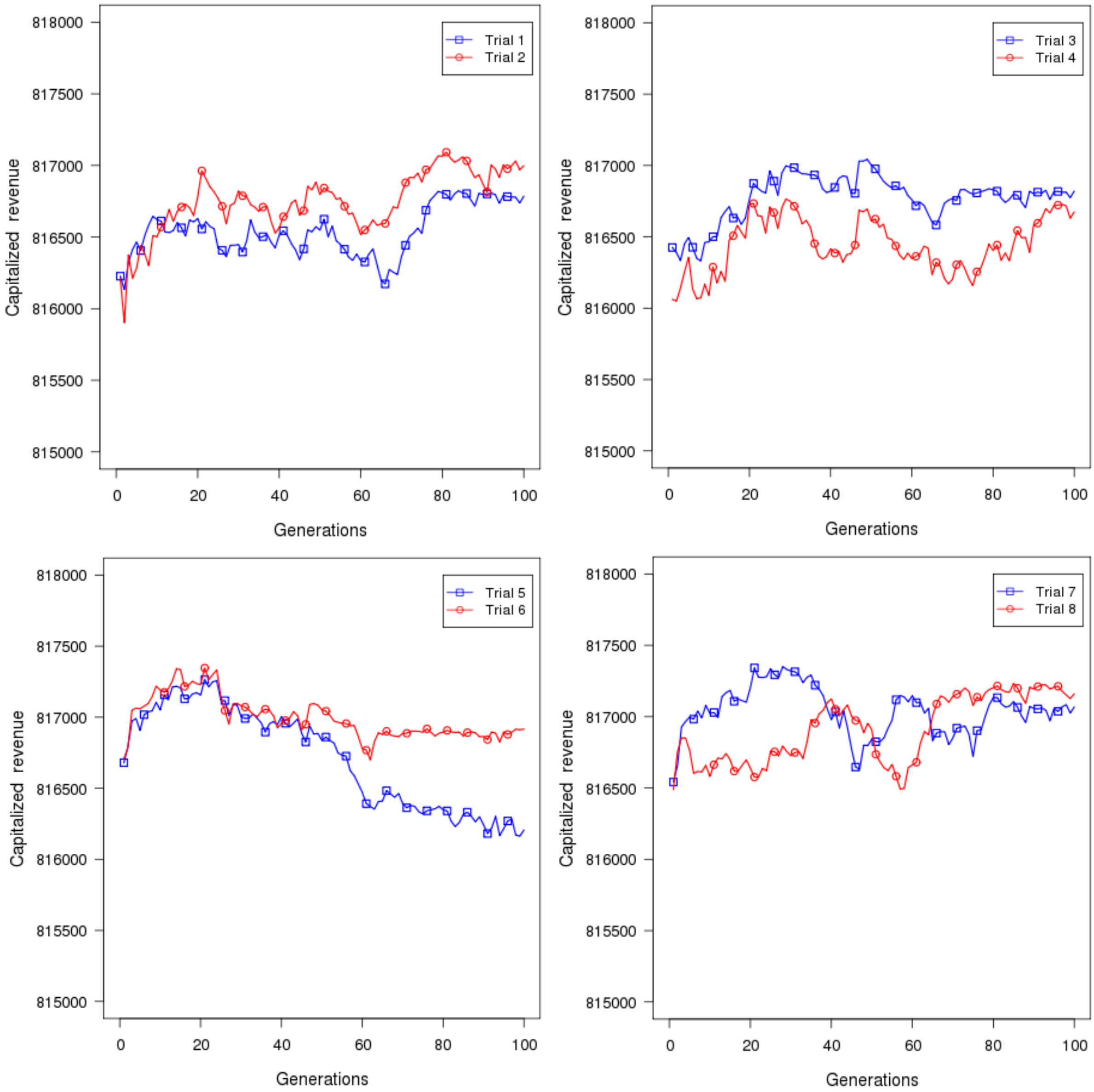


Figure 3.9: The generation wise evolution of capitalized revenue (in lakhs of Indian rupees) for different independent trial runs

The performance of the algorithm is analyzed by considering the generation wise evolution of the average fitness and the best fitness. The algorithm is able to improve values of the average fitness as well as the best fitness, towards the end of generations. The influence of chromosome elements on economic figures, like total cost and capitalized revenue, are also analyzed. The result shows that the chromosome elements have more influence on the total cost than the capitalized revenue. In essence, the study shows the suitability of application of standard GA, in the engineering design optimization problems like the one selected for the study i.e. the steam condenser optimization of a nuclear reactor.

CHAPTER 4

A STUDY ON GENETIC ALGORITHM METHODOLOGIES IN FUEL BUNDLE BURNUP OPTIMIZATION OF PRESSURIZED HEAVY WATER REACTOR

An optimization problem of nuclear fuel management with multiple objectives and constraints is considered in the chapter. The problem studied is the fuel bundle burnup optimization of Pressurized Heavy Water Reactor which optimizes the performance of the reactor core, while ensuring that operational and safety features are satisfied. As part of the study, two different GA methodologies, namely Penalty-function GA and Multi-objective GA, are applied and compared.

4.1. INTRODUCTION

Pressurized Heavy Water Reactors constitute the major category of commercial nuclear reactors, coming under the first stage of the three-stage nuclear power programme of India. The majority of operating nuclear power plants in India is based on Pressurized Heavy Water Reactor (PHWR) which can efficiently produce the fissile material required for the country's second stage of nuclear programme [17]. In India, there are eighteen such power plants in operation and four more plants of this category are under construction [135]. The optimization of fuel bundle burnup of PHWR involves in finding the arrangement of fresh and partially burned fuel bundles within the reactor core to optimize the performance of the reactor over the next operating cycle, while ensuring that operational constraints are always satisfied. The term 'burnup' defines the cumulative exposure of nuclear fuel in a reactor, which is expressed in terms of megawatt-days of thermal energy per metric ton (MWd/t) [15].

The aim of the study is to apply and evaluate the GA methodologies in deriving optimal discharge burnups which give maximum power, without violating various safety aspects of the reactor. The discharge burnups arrived by the GA based optimization procedure can be utilized in fixing the most suitable reference discharge burnups for the two burnup zones of the reactor core. Finding out the optimal values of reference discharge burnup of fuel bundles is important in maximizing the fuel utilization, while satisfying the safety and operational constraints of the reactor. The problem of the fuel bundle burnup optimization of a PHWR core has multiple objectives and constraints, some of which are in conflict with each other. This would result in the difficulty in optimization of all the parameters simultaneously. Therefore, optimization of the fuel bundle burnup is an involved task in terms of computational effort and time.

The reactor core considered in the present study is of an Indian PHWR (220 MWe) that uses natural uranium dioxide as fuel, heavy water as moderator and coolant. A brief description about the core of the reactor is given in the following section.

4.2. THE 220 MWe PHWR CORE

The PHWR core consists of a low-pressure horizontal reactor vessel ('calandria') containing the moderator at normal pressure and temperature. The calandria of the PHWR considered for this study is pierced by 306 pressure tubes, also known as coolant channels. Each coolant channel contains 12 cylindrical fuel bundles made up of zircaloy (an alloy of zirconium, tin and other metals), through which pressurized heavy water coolant circulates. Each fuel bundle holds 19 fuel elements and each fuel element consists of a stack of sintered cylindrical fuel pellets of natural uranium [136]. In PHWR, the fuelling is on-power (i.e. while reactor is in operation) and on a daily basis to continue the controlled chain reaction in the reactor. The fuelling operations are carried out by two remotely controlled fuelling machines, operating at each end of a fuel channel. The 8-bundle shift scheme is followed in the daily refueling operation in which eight fresh fuel bundles are inserted in to a channel from one end and eight burnt fuel bundles are taken out from the other end of the channel. The fuelling direction is opposite in adjacent channels. It helps in axial flux flattening [53]. The term “flux flattening” refers to a more uniform distribution of the power density across the reactor core which improves the average burnup of peripheral subassemblies [15].

The division of the core into two burnup zones is important for the PHWR. It is seen that without the burnup zones division, the radial neutron flux shape is peaked at the central region of the core and the bundle power at that region will exceed the operating limit at full power operation [53]. The radial flux flattening at the central region is essential to keep the maximum bundle power below the operating bundle power limit at full power operation. In order to get radial flux flattening, the reactor core is divided in to two burnup zones as shown in Figure 4.1. The inner zone contains 78 fuel channels and the outer zone contains 228 fuel channels. The typical discharge burnups for a 220 MWe PHWR are around 10200 MWd/t and 5500 MWd/t for inner and outer zones respectively. By keeping the discharge burnup of inner zone higher, the fissile inventory is kept relatively lower at the inner zone as compared to the outer zone channels. This leads to flattening of the neutron flux at the inner region and with that, the maximum bundle power is within operating bundle power limit [53].

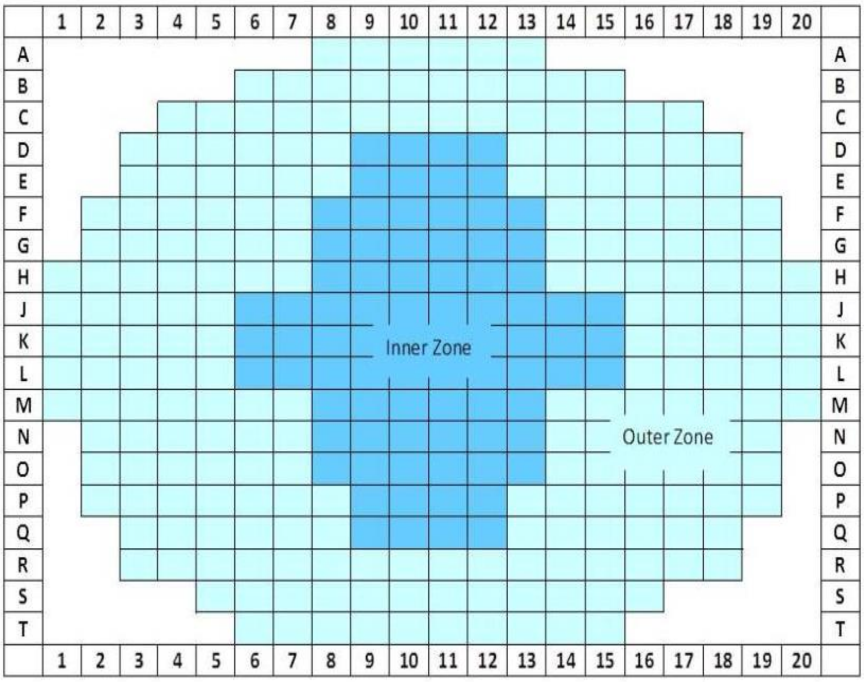


Figure 4.1: The two burnup zones of the 220 MWe Pressurized Heavy Water Reactor (PHWR) core

The channel coolant flows are fixed according to the reference channel powers in order to get uniform channel coolant outlet temperature. Since the channel flows are already fixed for the operating 220 MWe PHWR, the actual channel power cannot be kept more than its reference channel power. Therefore, maximum channel power comes as one of the constraints while deciding optimal discharge burnup for the inner and the outer zones of the reactor core.

The various parameters of the reactor core, required for the optimization procedure, are calculated by running the neutronics simulation code. The neutronics simulation code is used during the fitness evaluation step of the GA based optimization procedure. As has been seen during the discussion about the overall procedure of GA in nuclear fuel management (see Section 2.6 of Chapter 2), the neutronics simulation code plays a key role in predicting the reactor core behavior during the optimization procedure. The understanding about the basic functionalities and the overall working of neutronics simulation code is achieved by conducting several independent trial runs of the code. The details about the neutronics simulation code used in the study are described in the next section.

4.3. NEUTRONICS SIMULATION MODULE : TAQUIL

The reactor physics based simulation code used for this study is called TAQUIL which is developed in FORTRAN programming language [137, 138]. The code is particularly suitable for the simulation studies of equilibrium core of PHWRs. In TAQUIL, the refueling scheme is simulated using time-average model of the selected PHWR core. The presence of on-power refueling in PHWR results in the absence of a constant core power distribution shape for the equilibrium core. In the time-average model, channel wise power distribution is averaged over a period of fuel residence time to get approximately constant core power distribution. The time-average model is useful in determining three-dimensional power distribution, expected refueling frequency and the expected value of discharge burnup which produces maximum power without violating a set of safety and operational constraints. The discharge burnup values for the inner and outer zones act as input for TAQUIL code. The total core power and the refueling frequency are fixed; the code finds out bundle powers, channel powers, effective reactivity multiplication factors and average value of discharge burnup for the total core.

4.4. FUEL BUNDLE BURNUP OPTIMIZATION PROBLEM: MODEL FORMULATION

The aim of the optimization carried out in the present study is to find optimum values of reference discharge burnup for the inner and outer zones, in order to obtain maximum average discharge burnup for the total core. The values of discharge burnups arrived at can be utilized in fixing the most suitable reference discharge burnups for the two zones. Once the fuel bundle burnup reaches the corresponding zone's reference discharge burnup, then that fuel bundle can be replaced during the online refueling operation. In order to derive the appropriate values of reference discharge burnup through the optimization procedure, it is necessary to formulate a model that is suitable for applying different methodologies of GA.

During every iteration of GA (irrespective of the GA methodology considered), inner and outer burnup values are assigned to each of the chromosome of the GA population. The TAQUIL code is used to calculate corresponding neutronics parameter values. These output values from TAQUIL are used for the fitness evaluation stage in every iteration of GA. The passing of input burnup values from GA chromosomes to TAQUIL and parsing the required output values from the output file generated by TAQUIL to the GA are done by the interface module developed in the C programming language. The optimization problem of fuel bundle burnup is implemented with GA as the optimization tool and the neutronics simulation module (TAQUIL) as the fitness evaluation tool. The communication among these modules is smoothly achieved by the interface module.

This modular approach has been employed by following the GA based optimization procedure, described in Chapter 2 (Section 2.6). The mathematical model of the present optimization problem is incorporated in to the GA module. In the next section, the mathematical model formulation of the selected problem is considered.

4.4.1. The Mathematical Model of the Optimization Problem

The optimization problem taken for the study comes under the category of multi-objective optimization with four objectives and four constraints. The objectives/constraints are:

(i) maximum bundle power (*MBP*)

(ii) maximum channel power (*MCP*)

(iii) effective reactivity multiplication factor (K_{eff})

(iv) average discharge burnup of the total core (BU_{ave})

The "maximum bundle power" (*MBP*) and the "maximum channel power" (*MCP*) define limits to the maximum attainable power (by considering the safety and operational constraints) for the fuel bundles and channels respectively. The units of *MBP*, *MCP*, and BU_{ave} are kW, MW and MWd/t respectively. The primary objective for the study is the maximization of BU_{ave}. The minimization of *MBP* and *MCP* and the maximization of K_{eff} are the other objectives. This approach is followed to achieve the neutron flux-flattening requirement of the reactor core. As higher K_{eff} of the core leads to more excess reactivity of the core, which can be utilized for increasing the discharge burnup further, during the reactor operation. If the optimization study results in two burnup patterns, the one with higher K_{eff} , with all other parameters like BU_{ave} , *MBP* and *MCP* remains the same, then the pattern with higher K_{eff} is the better option. The present study considers the discharge burnup of inner zone as in the range between 8500 and 11000 MWd/t and that of outer zone as in the range between 4000 and 6500 MWd/t. A solution to the problem can be termed as feasible, only if it satisfies all the four constraints given. The optimization problem is defined with following objectives and constraints:

Objectives:

to maximize, BU_{ave} (of the total core)

to minimize, *MBP*

to minimize, *MCP*

to maximize, K_{eff}

Constraints:

MBP should be less than 430 kW

MCP should be less than 3.2 MW

K_{eff} should be greater than 1.0005

BU_{ave} should be greater than 6700 MWd/t

The given objectives are functions of the burnups of inner and outer zones. The mathematical formulation of the given problem is given as:

$$Max\ (BU_{ave}, K_{eff}) \text{ and } Min\ (MBP, MCP)$$

$$= f\,(\text{inner zone burnup, outer zone burnup}) \qquad (4.1)$$

Such that,

$$MBP < 430 \text{ KW}, MCP < 3.2 \text{ MW},$$

$$K_{eff} > 1.0005 \text{ and } BU_{ave} > 6700 \text{ MWd/t} \qquad (4.2)$$

where, *Max* represents the maximization, *Min* represents the minimization and f () represents “function of”. The problem has the following boundary conditions for the input values:

$$8500 \text{ MWd/t} \leq \text{inner zone burnup} \leq 11000 \text{ MWd/t} \qquad (4.3)$$

$$4000 \text{ MWd/t} \leq \text{outer zone burnup} \leq 6500 \text{ MWd/t} \qquad (4.4)$$

The mathematical model formulated above is a generic optimization model of the given problem. In order to apply the specific GA methodologies, the above model need to be further refined. The model formulations for the Penalty-function GA and for the Multi-objective GA are discussed in the following sections.

4.4.2. Optimization Model Formulation for the Penalty-function GA

In the case of Penalty-function GA, the multi-objective problem of fuel management optimization is converted in to single objective by adding penalty functions and constraints. A detailed description about the Penalty-function GA is given in Chapter 2 (Section 2.7.1). The model formulation of the Penalty-function GA for the selected optimization problem is considered next. Among the four objectives of the problem, the maximization of BU_{ave} is taken as the primary objective for Penalty-function GA and other three objectives are converted to penalty functions. The penalized objective function for the selected problem is formulated as follows:

$$Fitness = BU_{ave} - P1 - P2 - P3 \tag{4.5}$$

where,

$P1 = (MBP - MBP^0) \times Abp$

$P2 = (MCP - MCP^0) \times Acp$

$P3 = (K^0_{eff} - K_{eff}) \times Akef$

$P1$: penalty function related to maximum bundle power constraint

$P2$: penalty function related to maximum channel power constraint

$P3$: penalty function related to effective multiplication factor constraint

MBP^0 : maximum permitted bundle power (=430)

MCP^0 : maximum permitted channel power (=3.2)

K^0_{eff} : minimum required effective multiplication factor (=1.0005)

Fitness : penalized objective function used for the fitness evaluation of the GA.

The terms *Abp, Acp* and *Akef* are used to denote constant values selected to give proper weightage to penalty functions. These constants are estimated such that the penalty caused by each factor is numerically in the order as that of BU_{ave} (the magnitude of BU_{ave} is in the order of thousands). For the present work, the values of *Abp, Acp* and *Akef* are fixed as 10, 1000 and 100000 respectively. The penalty functions ($P1$, $P2$ and $P3$) are formulated in a special way for the present problem, as compared with the penalty approach commonly available in the literature [113, 118]. In the usual approach, penalty function is formulated in such a way that, it should not affect the actual objective function if constraints are not violated. On the other hand, in the case of constraint violation, the penalty function will put a high value in the opposite direction of the objective function. In the present work, the penalty function affects the objective function, even in the case of non-violation of the constraints (but in the same direction as that of the objective function). Therefore, the penalty functions additionally help in reaching the three objectives of the problem, i.e. *Max* (K_{eff}) and *Min* (*MBP*, *MCP*). In order to achieve this, the penalty functions are formulated such a way that, penalty function values (*P*1, *P*2 and *P*3) are becoming negative under the conditions when $MBP < MBP^0$ or $MCP < MCP^0$ or $K_{eff} > K^0_{eff}$.

4.4.3. Optimization Model Formulation for the Multi-objective GA

The implementation of Multi-objective GA followed is the Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) [116, 123]. A detailed description about the procedure of NSGA-II is covered in Chapter 2 (Section 2.7.4). The handling of the constraint violations in the NSGA-II is specific to the optimization problem being considered. The handling of constraint violations for the selected problem is explained next. As mentioned in Section 4.1, the optimization problem of fuel bundle burnup that considered is having four objectives and four constraints. The constraint violations are handled by an approach which is similar to the penalty handling mechanism of Penalty-function GA. The constraint functions are first normalized and then the violation for each constraint is calculated. For the four constraints of the selected problem, corresponding constraint violations are calculated as:

$$C1 = \frac{MBP - MBP^0}{MBP_{max} - MBP^0}, \quad \text{if } MBP > MBP^0$$
$$= 0, \quad \text{otherwise} \qquad (4.6)$$

$$C2 = \frac{MCP - MCP^0}{MCP_{max} - MCP^0}, \quad \text{if } MCP > MCP^0$$
$$= 0, \quad \text{otherwise} \qquad (4.7)$$

$$C3 = \frac{K^0{}_{eff} - K_{eff}}{K^0_{eff} - K_{effmin}}, \quad \text{if } K_{eff} < K^0_{eff}$$
$$= 0, \quad \text{otherwise} \qquad (4.8)$$

$$C4 = \frac{BU^0_{ave} - BU_{ave}}{BU^0_{ave} - BU_{avemin}}, \quad \text{if } BU_{ave} < BU^0_{ave}$$
$$= 0, \quad \text{otherwise} \qquad (4.9)$$

where,

$C1$: constraint violation value related to maximum bundle power

$C2$: constraint violation value related to maximum channel power

$C3$: constraint violation value related to effective multiplication factor

$C4$: constraint violation value related to average discharge burnup

MBP^0 : maximum permitted bundle power (=430)

MBP_{max}: maximum possible value of bundle power

MCP^0 : maximum permitted channel power (=3.2)

MCP_{max}: maximum possible value of channel power

K^0_{eff} : minimum required effective multiplication factor (=1.0005)

K_{effmin} : minimum possible value of effective multiplication factor

BU^0_{ave} : minimum limit for average discharge burnup (=6700)

BU_{avemin} : minimum possible value of average discharge burnup

The individual constraint violations corresponding to the four constraints of the problem are calculated (see Eqns. 4.6 to 4.9). Then the overall constraint violation (C_{tot}) is calculated as:

$$C_{tot} = C1+C2+C3+C4 \tag{4.10}$$

The next step is to modify each objective function value, according to the overall constraint violation. The overall constraint violation is multiplied with suitable constant values and the product is added to each of the objective function values to get the modified objective function values as:

$$MBP_{mod}=MBP + Bbp \times C_{tot} \tag{4.11}$$

$$MCP_{mod}=MCP + Bcp \times C_{tot} \tag{4.12}$$

$$K_{effmod}=Keff - Bkef \times C_{tot} \tag{4.13}$$

$$BU_{avemod}=BUave - Bbu \times C_{tot} \tag{4.14}$$

where,

MBP_{mod}	: modified value for the maximum bundle power
C_{tot}	: overall constraint violation
MCP_{mod}	: modified value for maximum channel power
K_{effmod}	: modified value for effective multiplication factor
BU_{avemod}	: modified value for average discharge burnup

The terms *Bbp, Bcp, Bkef* and *Bbu* are used to denote constant values selected to make both terms on the right side of the above equations to have the same order of magnitude. For a feasible solution, C_{tot} is calculated as 0 and the modified values of the objective functions are same as that of actual objective function values. For an infeasible solution, a penalty is added to each of the objective function corresponding to overall constraint violation. Once the modified objective functions are calculated, those values are used by the algorithm for Pareto-optimal fronts sorting. The Pareto-optimal front sorting is followed based on non-domination ranks and crowded comparison operation in the NSGA-II implementation, as described in Chapter 2 (Section 2.7.4).

4.5. DETAILS OF IMPLEMENTATION

The real-number representation (see Chapter 2, Section 2.3.2) is selected for the implementations of both Penalty-function GA and Multi-objective GA. The real-number representation offers a natural way of chromosome representation without explicit encoding mechanism, when the decision variables are real numbers [36, 113]. In the present study, the decision variables i.e. inner and outer zone burnup values, are real numbers and can be directly encoded as chromosome. Therefore, the real-number representation has been selected. The 'C' programming language is used for implementation of the Penalty-function GA and the Multi-objective GA. The interface module for the communication between the GA modules and the neutronics simulation module (i.e. TAQUIL) is also developed in the 'C' programming language. While comparing the performance of the two algorithms, it is important to keep the GA related parameters, corresponding values and methods as the same. The information relating to the GA parameters, employed in the optimization procedure, is given in the Table 4.1. The methods and values given in the Table 4.1 are common for the Penalty-function GA and the Multi-objective GA.

The first step of the algorithm procedure is to generate the initial parent population. Each element in the chromosome vector is assigned to be within the desired range at the time of generation of the initial population. The crossover and mutation operators selected for the GA implementations are also designed to meet this requirement.

Table 4.1: Genetic Parameters and methods or values used in the GA based optimization procedure. The values given in the table are the common parameters for the Penalty-function GA and the Multi-objective GA.

Parameter	**Methods/Values**
Encoding	Floating point
Population size	50
Crossover Method	Arithmetical
Crossover Probability (CR)	0.6
Mutation Method	Non-uniform
Mutation Probability (MR)	0.05
Maximum no. of Generations	100

As an example, two typical chromosomes vectors, x_1 and x_2 generated by the algorithm can be represented as:

$$x_1 = (10281.35\,, 6034.42) \qquad (4.15)$$

$$x_2 = (8941.24\,, 5281.71) \qquad (4.16)$$

The first elements in x_1 and x_2 denote the inner zone burnup values and the second elements denote the outer zone burnup values. As has been seen in Chapter 2, the Arithmetical crossover is a method suitable for real-number representation and it is followed in the study. For the chromosomes vectors considered in the above example, i.e. x_1 = (10281.35, 6034.42) and x_2 = (8941.24, 5281.71), the resulting offspring, o_1 and o_2, after arithmetical crossover operation is represented as:

$$o_1 = \left(\mathrm{a} \times 10281.35 + (1 - \mathrm{a}) \times 8941.24\,, \right.$$

$$\left. \mathrm{a} \times 6034.42 + (1 - \mathrm{a}) \times 5281.71 \right) \qquad (4.17)$$

$$o_2 = \left(\mathrm{a} \times 8941.24 + (1 - \mathrm{a}) \times 10281.35\,, \right.$$

$$\left. \mathrm{a} \times 5281.71 + (1 - \mathrm{a}) \times 6034.42 \right) \qquad (4.18)$$

(by following the equations which define the arithmetical crossover, given in Chapter 2, i.e. Eqns. 2.2 and 2.3)

At a particular instance, when a = 0.3351 (the random number generated between 0 and 1), the resultant offspring chromosomes are:

$$o_1 = (9390.31, 5533.94) \qquad (4.19)$$

$$o_2 = (9832.28, 5782.19) \qquad (4.20)$$

The mutation method followed is the non-uniform mutation for both Penalty-function GA and Multi-objective GA. The details about the non-uniform mutation are given in Chapter 2 (Section 2.4.8.). In continuation of the example considered above, assuming that one of the offspring (o_1) derived from crossover operation is undergoing the non-uniform mutation at 40th generation (t=40). If the second element of o_1 (that represents the outer zone burnup) is chosen for the mutation operation (i.e. $v_k = 5782.19$). The *UB* and *LB* values, representing the upper and lower bounds for the outer zone burnup, are 6500.00 and 4000.00 respectively. If the random digit generated as part of non-uniform mutation operator is 0 (by selecting either 0 or 1 with equal probability), then the mutated element v_k^l can be calculated as:

$$v_k^l = 5782.19 + \Delta\,[40, (6500.00 - 5782.19)] \qquad (4.21)$$

(by following the equations which define the non-uniform mutation, given in Chapter 2, i.e. Eqn. 2.4 and 2.5)

Now, if $r = 0.70$, then $\Delta\,(t, y)$ function in the above step will return 19.63 and v_k^l will get 5801.82. In essence, for the considered example, the non-uniform mutation operation changes the outer burnup value from 5782.19 to 5801.82.

The methods followed in the study for producing the new population are considered next. The initial population is generated by calling the random number generation function of the C programming language. Using the function, random numbers are generated within the range of input boundary conditions. The selection used in Penalty-function GA is Roulette-Wheel method, which is discussed in Chapter 2 (Section 2.4.2.). The selection in Multi-objective GA is based on the ranks assigned to individual chromosomes by non-dominated sorting of Pareto-optimal fronts, as discussed in Chapter 2 (Section 2.7.4.). The concept of *Elitism* is also incorporated in the selection method of Penalty-function GA, where a fixed number the chromosomes having higher fitness values are considered as elite chromosomes and are retained in the new generation. The number of elite chromosomes in each generation is assigned as five for the implementation of Penalty-function GA.

4.6. RESULTS

Several trial runs were conducted with randomly generated initial GA population (both for the Penalty Function-GA and for the Multi-objective GA). Each trial run was started with entirely different initial populations, ensuring different initial search space for different trial runs. Based on the results generated initially, GA parameters and penalty coefficients were fine-tuned. Both GA methodologies were studied in detail to ascertain the suitability for the selected problem. During the comparison process: population size, maximum number of generations, encoding scheme of GA, crossover mutation methods and their probability values, are kept same for both the algorithms and are given in Table 4.1. The fine-tuned versions of the algorithms are used to generate final results by conducting twenty trial runs for each of the algorithms.

4.6.1. Comparison of Maximum and Minimum Values of Objective Functions

The maximum and minimum values of the four objective functions, obtained in the final generation (i.e., 100th generation in the present study) of the Penalty Function-GA and the Multi-objective GA, are given in Tables 4.2 and 4.3., respectively. The maximum and minimum values of the solutions arrived at (i.e. burnup values of inner and outer zones) are also presented. The feasible solutions generated by both the algorithms are considered. The TAQUIL code can accept inputs and generate outputs with accuracy up to six decimal places. The results presented for burnups and *MBP* are rounded to two decimal places. In the case of *MCP*, the selected values were rounded off to five decimal places and that of K_{eff} to six decimal places, to reflect the variation in the expected range. It can be seen from the results that the maximum and the minimum values obtained for all the objective functions by the Penalty Function-GA lies in narrower range as compared to Multi-objective GA.

Table 4.2: Maximum and minimum values of the four objective functions and the solutions arrived in the final generation of the Penalty Function-GA (results of twenty independent trial runs are presented)

Trial No.	Max Min	Values obtained for objective function				Solution arrived	
		MBP	*MCP*	K_{eff}	BU_{ave}	Inner zone burn up	Outer zone burn up
1	Max	423.27	3.09439	1.000503	6835.85	10416.73	5954.35
	Min	423.23	3.09416	1.000502	6835.81	10416.46	5954.32
2	Max	423.25	3.09429	1.000506	6835.53	10416.49	5953.95
	Min	423.23	3.09418	1.000504	6835.2	10416.19	5953.66
3	Max	423.99	3.09955	1.000563	6834.57	10393.71	5954.95
	Min	423.94	3.09916	1.000558	6834.02	10392.57	5954.50
4	Max	423.42	3.09549	1.000521	6834.76	10412.51	5953.69
	Min	423.33	3.09487	1.000516	6834.6	10410.43	5953.45
5	Max	423.27	3.09436	1.000522	6833.19	10413.98	5951.82
	Min	423.25	3.09425	1.000519	6832.82	10413.84	5951.44
6	Max	423.27	3.09447	1.000507	6835.75	10415.87	5954.29
	Min	423.25	3.09434	1.000504	6835.26	10415.36	5953.85
7	Max	423.28	3.09399	1.000506	6835.24	10417.66	5953.59
	Min	423.23	3.09367	1.000503	6834.54	10417.43	5952.89
8	Max	423.28	3.09451	1.000527	6834.10	10416.49	5952.49
	Min	423.22	3.09390	1.000510	6832.13	10412.51	5950.85
9	Max	423.30	3.09472	1.000510	6836.52	10415.96	5955.06
	Min	423.27	3.09443	1.000501	6835.25	10414.48	5953.87
10	Max	423.26	3.09434	1.000505	6835.96	10416.82	5954.42
	Min	423.23	3.09410	1.000501	6835.25	10416.65	5953.74
11	Max	423.27	3.09415	1.000502	6835.84	10417.62	5954.17
	Min	423.22	3.09394	1.000500	6835.65	10417.09	5954.13
12	Max	423.27	3.09408	1.000515	6833.99	10416.14	5952.42
	Min	423.22	3.09390	1.000511	6833.32	10415.52	5951.85
13	Max	423.30	3.09470	1.000506	6836.30	10415.67	5954.85
	Min	423.27	3.09448	1.000502	6835.95	10414.68	5954.59
14	Max	423.27	3.09438	1.000504	6836.15	10416.84	5954.58
	Min	423.24	3.09420	1.000501	6835.57	10416.06	5954.09
15	Max	423.42	3.09549	1.000524	6834.19	10410.48	5953.12
	Min	423.38	3.09525	1.000522	6834.13	10410.19	5953.1
16	Max	423.27	3.09417	1.000507	6835.15	10417.24	5953.5
	Min	423.21	3.09388	1.000504	6834.69	10416.87	5953.12
17	Max	423.27	3.09428	1.000520	6834.46	10415.97	5952.9
	Min	423.23	3.09403	1.000510	6832.94	10414.36	5951.49
18	Max	423.36	3.09513	1.000503	6837.21	10414.75	5955.89
	Min	423.33	3.09487	1.000501	6836.8	10414.34	5955.53
19	Max	423.29	3.09443	1.000545	6829.72	10411.11	5948.43
	Min	423.25	3.09424	1.000540	6829.01	10410.29	5947.83
20	Max	423.27	3.09416	1.000503	6835.9	10417.43	5954.28
	Min	423.23	3.09398	1.000501	6835.56	10416.9	5953.99

Table 4.3: Maximum and minimum values of the four objective functions and the solutions arrived in the final generation of the Multi-objective GA (results of twenty independent trial runs are presented)

Trial No.	Max Min	Values obtained for objective functions				Solutions arrived	
		MBP	*MCP*	K_{eff}	BU_{ave}	Inner zone burn up	Outer zone burn up
1	Max	429.44	3.13845	1.001593	6900.34	10440.74	6033.63
	Min	423.23	3.07084	1.000504	6706.74	10170.42	5841.56
2	Max	430.00	3.14217	1.001548	6905.55	10426.05	6039.38
	Min	423.31	3.07111	1.000501	6700.83	10193.47	5831.75
3	Max	429.99	3.14166	1.001723	6904.63	10467.83	6038.51
	Min	423.25	3.07096	1.000501	6702.62	10108.28	5831.70
4	Max	429.84	3.13966	1.001698	6886.37	10445.63	6018.79
	Min	423.28	3.07105	1.000512	6705.40	10114.87	5833.03
5	Max	429.72	3.13972	1.001568	6896.90	10416.81	6028.71
	Min	423.21	3.07113	1.000500	6703.16	10165.89	5818.01
6	Max	430.00	3.14193	1.001498	6885.35	10437.8	6017.18
	Min	423.25	3.07077	1.000545	6700.10	10216.98	5818.17
7	Max	429.68	3.14015	1.001411	6900.35	10488.18	6034.25
	Min	423.23	3.07092	1.000500	6705.43	10220.24	5829.27
8	Max	429.93	3.14164	1.001595	6903.31	10462.43	6037.57
	Min	423.29	3.07095	1.000503	6712.12	10135.4	5841.25
9	Max	429.88	3.14152	1.001707	6897.55	10466.59	6032.05
	Min	423.26	3.07115	1.000509	6705.36	10121.83	5844.49
10	Max	429.98	3.14210	1.001509	6902.74	10423.73	6036.68
	Min	423.23	3.07104	1.000511	6700.27	10153.91	5821.33
11	Max	429.96	3.14172	1.001557	6886.05	10198.87	6016.54
	Min	423.27	3.07096	1.000504	6700.21	10455.82	5819.02
12	Max	429.97	3.14222	1.001580	6899.81	10468.28	6034.63
	Min	423.25	3.07110	1.000519	6700.28	10184.99	5825.25
13	Max	430.00	3.14181	1.001598	6905.85	10460.55	6040.07
	Min	423.25	3.07096	1.000509	6703.83	10129.58	5830.82
14	Max	429.79	3.14094	1.001546	6906.33	10453.21	6040.38
	Min	423.23	3.07096	1.000502	6702.29	10151.93	5820.18
15	Max	430.00	3.14192	1.001602	6898.35	10447.46	6030.53
	Min	423.21	3.07105	1.000501	6702.68	10155.7	5823.34
16	Max	429.97	3.14117	1.001497	6878.56	10446.75	6006.08
	Min	423.24	3.07115	1.000500	6700.01	10163.55	5829.69
17	Max	429.96	3.14194	1.001318	6894.81	10455.6	6026.16
	Min	423.24	3.07085	1.000502	6702.76	10248.57	5826.06
18	Max	429.99	3.14184	1.001587	6906.15	10448.45	6040.09
	Min	423.27	3.07098	1.000501	6700.59	10160.15	5823.10
19	Max	429.89	3.14163	1.001655	6905.12	10426.97	6039.46
	Min	423.28	3.07101	1.000515	6700.24	10154.8	5836.22
20	Max	430.00	3.14244	1.001594	6907.53	10462.62	6041.93
	Min	423.23	3.07102	1.000502	6700.83	10151.43	5818.21

The maximum and minimum values of the feasible solutions arrived at, also show the same behavior. This behavior is consistent with the results of twenty trial runs of the algorithms. The major observations from the results are:

- Both, Penalty Function-GA and Multi-objective GA, are able to produce feasible solutions at the final generation.
- Always, the Multi-objective GA is capable of generating wide range of feasible values for the objective functions and for the solutions.

The ability of Multi-objective GA to generate wide range of different feasible solutions is an important feature for the selected nuclear fuel management problem. That helps the reactor operator in getting more choices when deciding the appropriate discharge burnup of the core zones.

4.6.2. Comparison of GA Performance: Generation wise Production of Feasible Solutions

The number of feasible or good solutions produced in successive generations can be treated as a measure of effectiveness of GA implementation [36, 113]. This indicates a measure of "speed of convergence" of the algorithm. For the selected problem of fuel bundle optimization, one generated solution is termed as feasible, if it satisfies all the four constraints. That is, a feasible solution satisfies the conditions such as, *MBP* should be less than 430 KW, *MCP* should be less than 3.2 MW, K_{eff} should be greater than 1.0005 and BU_{ave} should be greater than 6700 MWd/t.

The average numbers of feasible solutions produced in successive generations by the two algorithms are compared in Figure 4.2. The average number of feasible solutions produced by considering the twenty trial runs of the Penalty-function GA is given in Figure 4.2(a) and that of the Multi-objective GA is given in Figure 4.2(b).

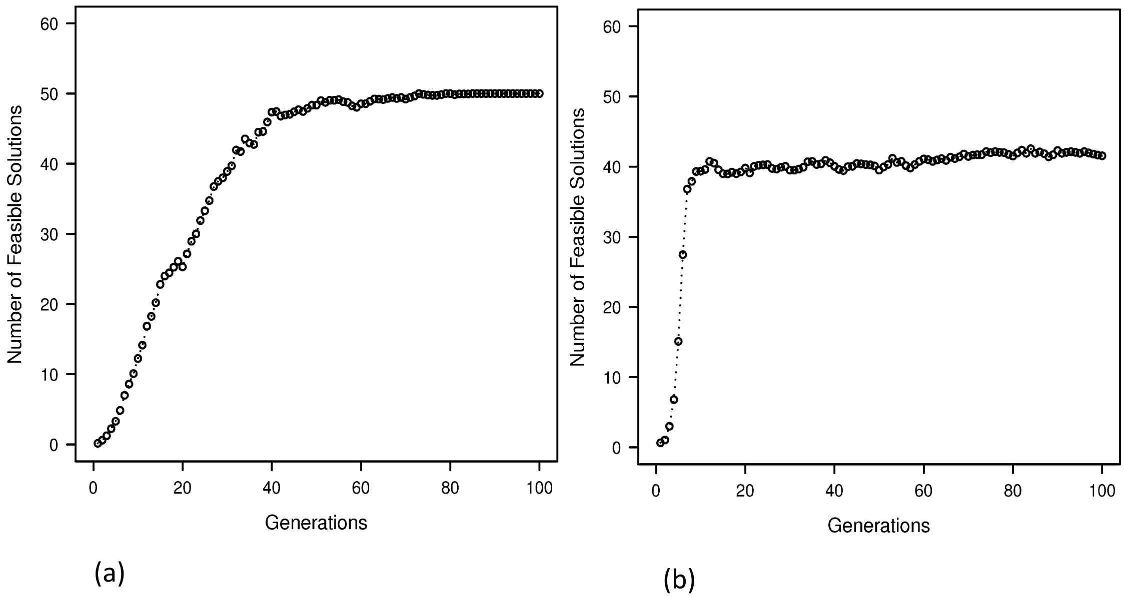


Figure 4.2: Average number of feasible solutions (generation wise) for twenty trial runs: (a) for Penalty-function GA. (b) for Multi-objective GA.

The conditions of feasibility are kept the same for both the algorithms. The Multi-objective GA produces feasible solutions in a faster rate at earlier generations and has a better convergence speed. A comparison is made on GA performance based on the average CPU time taken to produce equal number of number of feasible solutions. The comparison is done on a computer system with Dual Six Core 64 bit Intel Xeon @ 3.06 GHz processor and 48 GB RAM. The average CPU times and generations taken for producing 80% of population with feasible solutions, (i.e. 40 feasible solutions out of the 50 members of the population) are taken in the comparison. The Penalty-function GA took 32 generations and 174.82 seconds to produce 40 feasible solutions. The Multi-objective GA took 12 generations and 60.04 seconds to produce the same number of 40 feasible solutions; implying that the Multi-objective GA is 66% faster than Penalty-function GA, with respect to CPU time for generating 80% of the population with feasible solutions. At the same time, the number of generations taken (for generating 40 feasible solutions) by Multi-objective GA is 63%

less than the number needed for the Penalty-function GA. The behavior observed can be utilized in reducing the computational time by lowering the number of required generations of the algorithm. Another observation is that the when the computational time requirement for a fixed number of generations are considered, there is no significant difference between Penalty-function and Multi-objective GA. The average CPU time requirement (for twenty trial runs, each with 100 generations) for Penalty-function GA is 546.31 seconds whereas for Multi-objective GA it is 500.36 seconds. It is seen that, when average computational time of fixed generations are considered, Multi-objective GA has a marginal advantage of 8% faster than Penalty-function GA. The overall observation from the comparison is that the Multi-objective GA has a significant advantage in convergence speed and has a marginal advantage in computational time, when compared with the Penalty-function GA.

4.6.3. Convergence of Objective functions: Final Generation

The feasible solutions generated at the final generations of Penalty-function GA and Multi-objective GA are considered in the comparison. The results generated by four trial runs are presented in Figure 4.3. The distributions of generated objective function values in feasible solutions, at the final generation (i.e. 100th generation) are effectively represented by box plots (also known as box and whisker plots) [139]. The bottom and top of the box represent the first and third quartiles (Q1 and Q3) and the band inside the box represents the second quartile or the median (Q2). The vertical dotted lines at each end of the box are called whiskers. The bottom whisker goes from first quartile to the smallest non-outlier (outlier represents data point that diverges greatly from the overall pattern of data) in the data set; the top whisker goes from third quartile to the largest non-outlier. The outliers are represented by small circles (for our results, there is no outliers and hence no circles shown in Figure 4.3). The inter-quartile range, which is a measure of variability, is represented by the vertical height of the box (i.e. Q3 minus Q1). The distributions of objective functions for four trial runs of Penalty-function GA are plotted in left hand side and those for Multi-objective GA are plotted in the right hand side in the figure (Figure 4.3). The results for the two algorithms are plotted under the same scale, which enables a good comparison. The vertical heights of the boxes represent the spread of feasible solutions for 50% of the data samples. The boxes generated for the Penalty-function GA are so narrow that they are represented as simple thick lines.

The feasible solutions generated at the final generations of all the twenty trial runs for each of the algorithms are considered next. The maximum (Max), minimum (Min), average (Ave) and standard deviation (SD) values for the four objective functions are given in Table 4.4. The maximum values are calculated by taking the average of maximum values produced (separately for each of the four objective functions) at the final generation in twenty trial runs. Similarly, the minimum values are calculated by taking the average of minimum values generated. The average values shown in the table are calculated by finding the average of respective objective function values (by considering the 50 members of the final population) for twenty trial runs and corresponding standard deviations are also calculated.

Following observations can be made from the comparison of the distribution of feasible solutions in final generation given in Figure 4.3 and Table 4.4:

- Multi-objective GA produces wide range of feasible solutions in the final generation with respect to the four objectives of the selected problem.
- Penalty-function GA produces feasible solutions in much narrow range in the final generation.

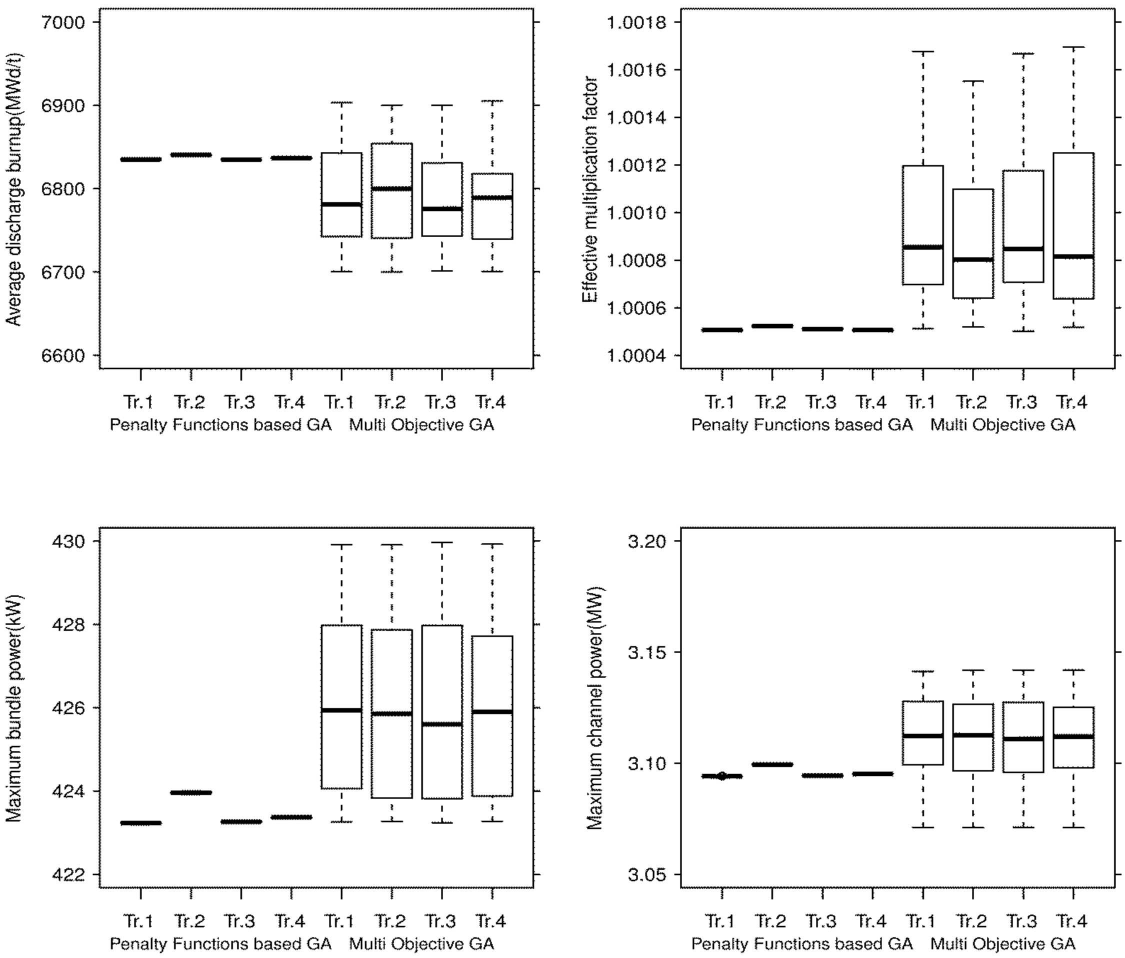


Figure 4.3: Box plots to represent the four objective functions distribution among the generated feasible solutions at the final generation (for four trial runs). A comparison between Penalty-function GA and Multi-objective GA is given in every plot.

Table 4.4: Feasible Solution values obtained in final generation considering the twenty trial runs.

	Penalty Function GA				Multi-objective GA			
	MBP	*MCP*	K_{eff}	BU_{ave}	*MBP*	*MCP*	K_{eff}	BU_{ave}
Max	423.33	3.0948	1.000515	6835.02	429.90	3.1413	1.001569	6898.58
Min	423.29	3.0945	1.000511	6834.43	423.25	3.0710	1.000507	6702.79
Ave	423.30	3.0946	1.000512	6834.83	426.12	3.1112	1.000897	6797.50
SD	0.1600	0.0012	$1.5x10^{-05}$	1.6100	0.2900	0.0024	$5.1x10^{-05}$	8.5200

4.6.4. Generation wise evolution of Objective functions

The convergence of the algorithms at the end of generation has been discussed above. Now, the convergence of the given objectives during the whole evolution of the Penalty-function GA and Multi-objective GA is considered. The results are plotted in Figure 4.4. The solid black points denote the average value of the objective for the whole population of that particular generation. The dotted lines above and below denote the standard deviation.

The convergences of the four objectives for Penalty-function GA are shown in Figure 4.4(a) to 4.4(d). Similarly, the convergences of the four objectives of Multi-objective GA are shown Figure 4.4(e) to 4.4(h). Following are the observations from the comparison of the convergences of the objectives:

- Both Penalty-function GA and Multi-objective GA are capable of arriving at around the same converged area with respect to the given objectives.
- The objective functions have converged to a narrower region for Penalty-function GA as compared to Multi-objective GA.
- The speed of convergence is faster for Multi-objective GA as compared to Penalty-function GA.

4.6.5. Aggregate Diversity of Objective Functions: Generation wise

The term ‘diversity’ represents the distribution of the selected parameter within the allowable range. The aim is to consider the generation wise evolution of Penalty-function GA and Multi-objective GA; by comparing the aggregate diversity of all the four objective functions. The diversity in the values of these objective functions, gives further insight about the range of generated final objective values.

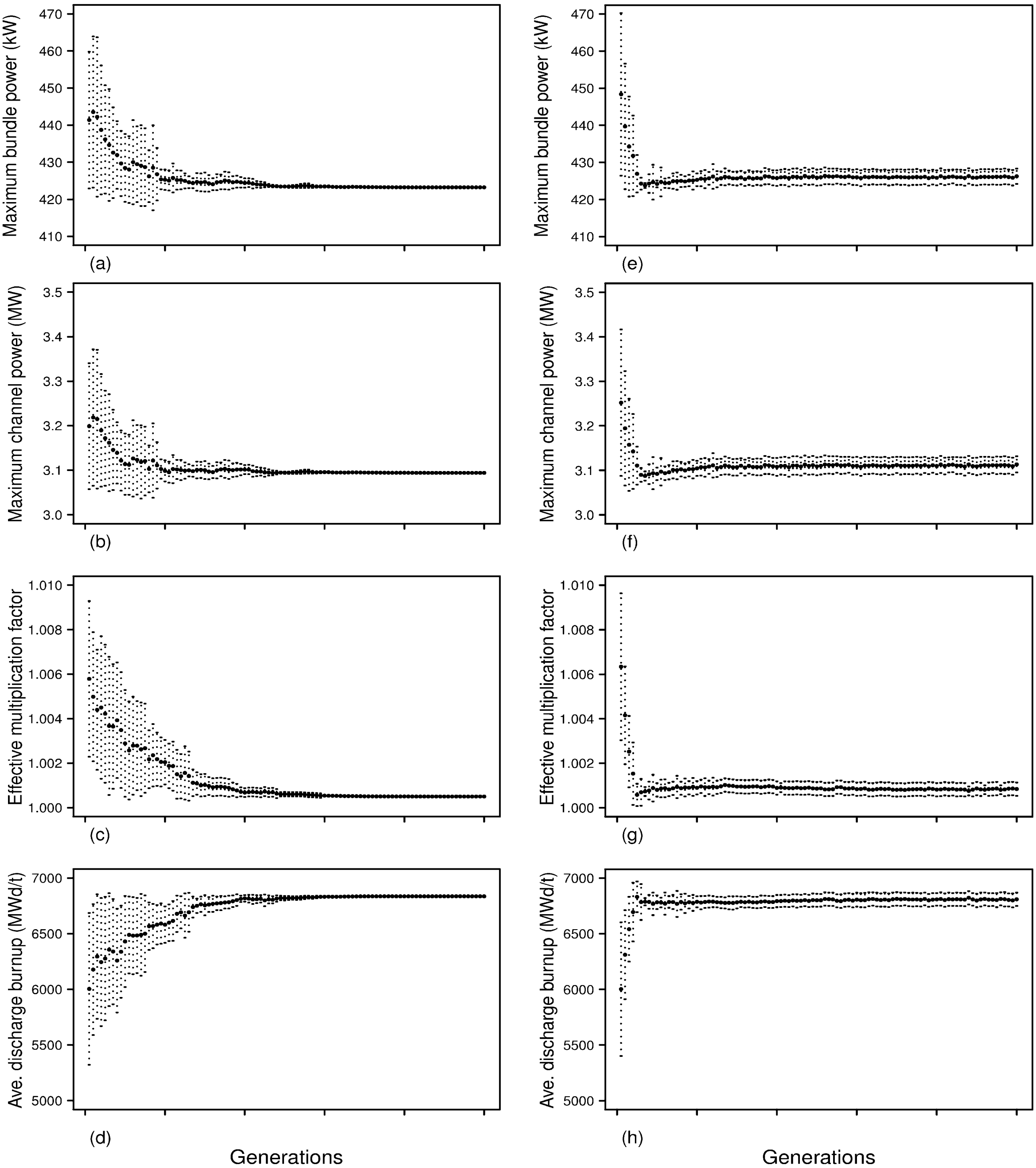


Figure 4.4: Convergences of the objectives during the generation wise evolution of the algorithms: (a) to (d) for Penalty-function GA. (e) to (h) for Multi-objective GA.

A common procedure is followed to find the aggregate diversity of four objective functions. The aggregate diversity of generated solutions, at successive stages of generations is calculated using the following procedure:

- The arithmetic mean is calculated for each objective function in the population (The population size is 50 and each individual of the population represents a solution set with four objective function values).
- The distance from the arithmetic mean is calculated for each individual's objective functions and square the distances.
- Summation of the distances (for every objective function separately) for all the individuals of the population.
- Dividing the total distance found for every objective function by the population size. (Corresponding to the four objective functions, four values are thus obtained).
- Normalize the four values obtained in the previous step, on a 0 to 10 scale.
- Calculate the average of the normalized values which represents the aggregate diversity of the four objective functions for that generation.

Using the above procedure, the normalized aggregate diversity is calculated for all successive generations for both of the algorithms, i.e. Penalty-function GA and Multi-objective GA. The complete diversity plots (i.e. for all generations from first generation to hundredth generation) for a trial run of Penalty-function GA and Multi-objective GA are shown in Figure 4.5.

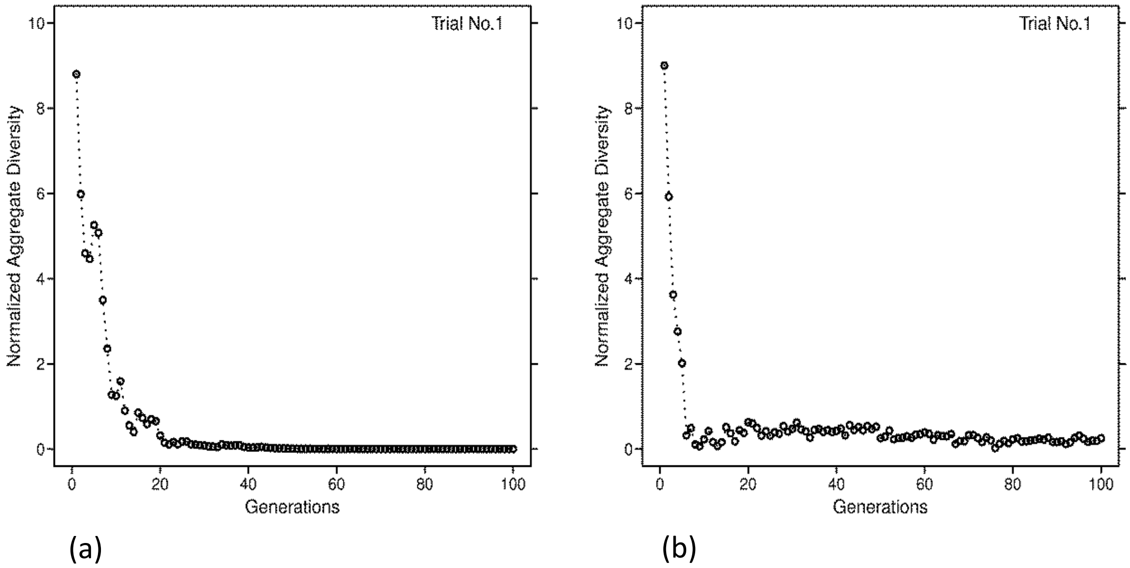


Figure 4.5: Generation wise evolution of normalized aggregate diversity of four objective functions (a) for Penalty-function GA. (b) for Multi-objective GA. (from first generation to hundredth generation)

Wide ranges of values of objective functions are produced in both of the algorithms' initial generations. The reason for this is that both algorithms are started from the randomly generated initial solution space which has the maximum diversity. The range of diversity narrows down towards the end generations. The variation in diversity is approaching a very small value (near zero in a normalized scale between 0 and 10) for Penalty-function GA but it is still better for Multi-objective GA (see Figure 4.5).

The diversity towards end generations can be represented in a wider scale, if a few initial generations are removed from the plots. The first nine generations are omitted (to represent the diversity at the end generations in a wider scale) and the result plots are given in Figures 4.6 and 4.7. The results generated by four trial runs (randomly selected from the twenty trial runs) are presented here. The normalized aggregate diversity over the evolution from tenth generation up to hundredth generation, for the Penalty-function GA is shown in Figure 4.6. It can be seen that for the Penalty-function GA, diversity among the objective functions is drastically reducing and approaching a very small value (near zero) towards the end generations. This clearly indicates that a high number of solutions are generated within a very small segment of the entire solution range by the Penalty-function GA. Similarly, Figure 4.7 shows the normalized aggregate diversity over the evolution from tenth generation up to hundredth generation for the Multi-objective GA. The results show the ability of Multi-objective GA to retain the diversity among the objective functions, in a much better way as compared to Penalty-function GA. Thus, Multi-objective GA produces wide range feasible solutions as final results.

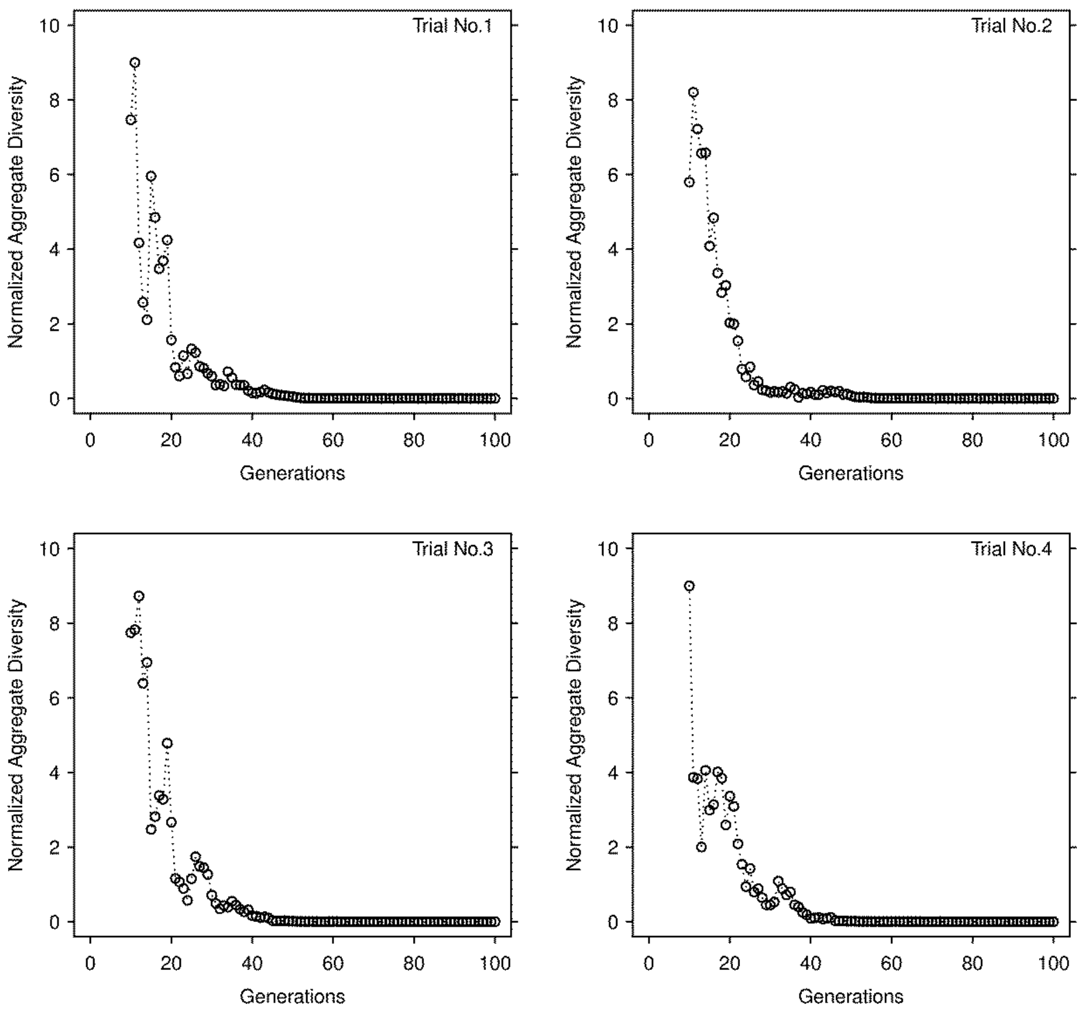


Figure 4.6: Generation wise evolution of normalized aggregate diversity of four objective functions for Penalty-function GA (from tenth generation to hundredth generation)

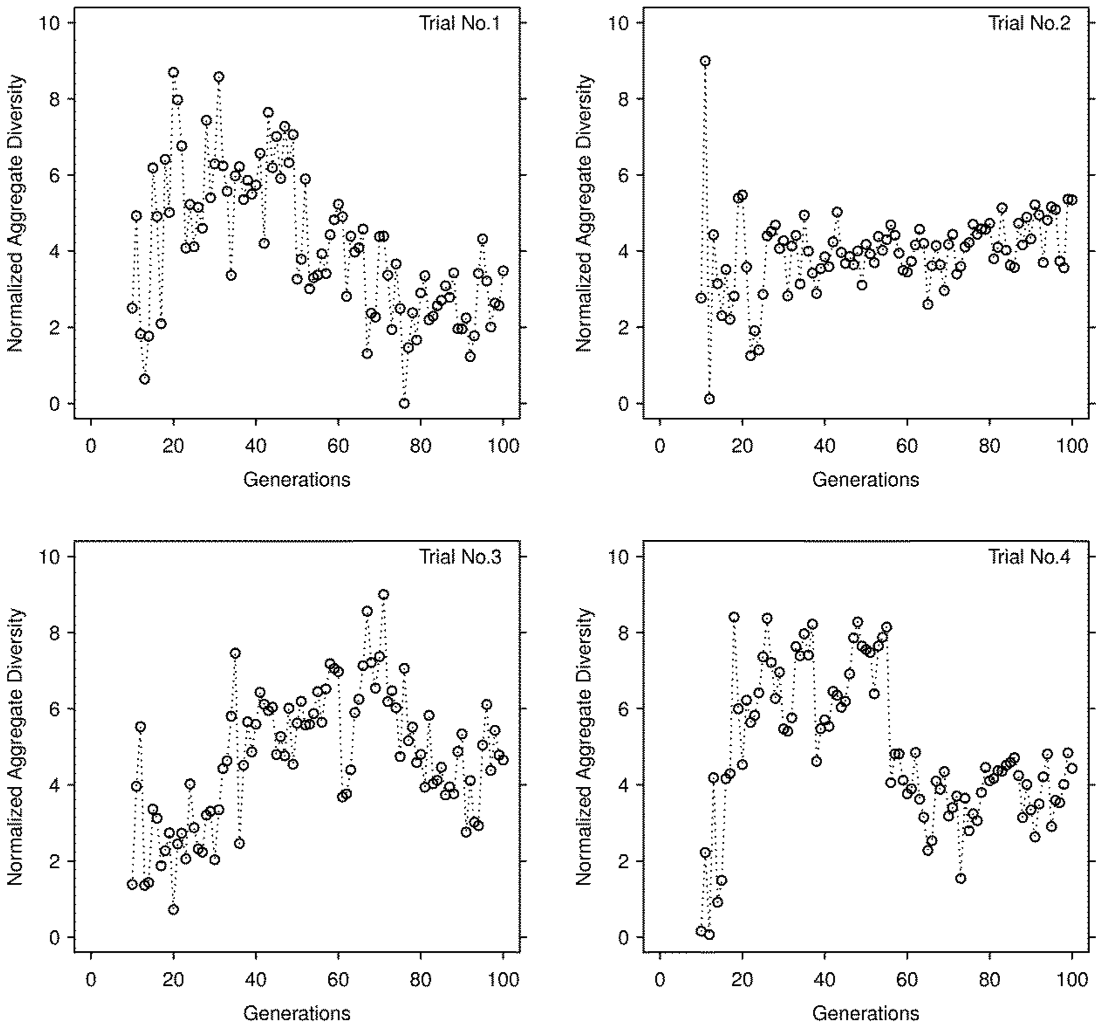


Figure 4.7: Generation wise evolution of normalized aggregate diversity of four objective functions for Multi-objective GA (from tenth generation to hundredth generation)

Following observations can be made from the comparison of the normalized aggregate diversity plots given in Figures 4.6 and 4.7:

- Aggregate diversity among the four objective function values are shrinking to a narrow range (near zero) towards the end of generations for the Penalty based GA.
- On the other hand, Multi-objective GA is capable of retaining the wider range of solutions with respect to all the four objectives of the selected problem.

4.6.6. Solution Diversity: Initial Population Vs. Final Population

The inner and outer zone burnup values forms the solution candidates for the selected problem. The distribution of solution candidates in the initial generation population and final generation population (in the overall search space) are compared in Figure 4.8. The Figure 4.8(a) shows solution diversity of the initial and the final populations for Penalty-function GA and Figure 4.8(b) show that of Multi-objective GA.

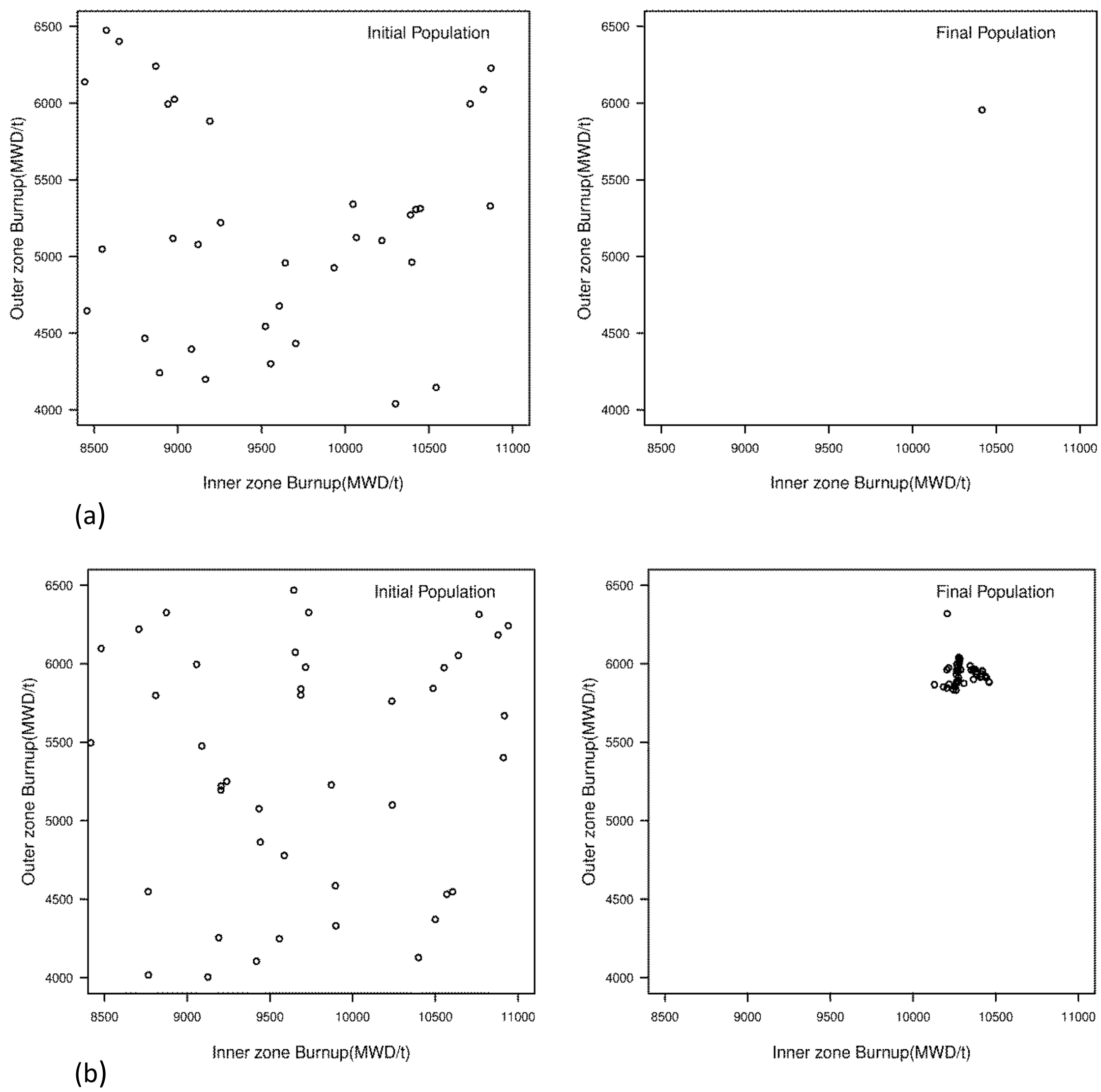


Figure 4.8: Solution diversity comparison of the algorithms: (a) Penalty-function GA: initial population vs. final population (b) Multi-objective GA: initial population vs. final population

The total population and corresponding solution candidates of a single trial run are being considered here. The solution values converge to a very narrow area (represented as single point in the diagram) in the case of Penalty-function GA, but to a wider area for Multi-objective GA. This shows that, as in the case of diversity of objective functions, the solution diversity also is better for Multi-objective GA.

4.7. SUMMARY

In this study, two different GA methodologies namely Penalty-function GA and Multi-objective GA are applied in the fuel bundle burnup optimization of PHWR. The result obtained from the study shows the suitability of Multi-objective GA over Penalty-function GA, in solving problems like the one selected for this study. The Multi-objective GA is able to generate diverse optimal solutions with respect to all the five objectives considered in this study. The ability to find much better spread of solutions by the Multi-objective GA is a key point with respect to the present study. It is resulting in getting more choices for the reactor operator while deciding the fuel bundle discharge burnups of inner and outer zones of the reactor core. Another behavior shown by the Multi-objective GA for the selected problem is the better speed of convergence compared with the Penalty-function GA. This implies that the total number of generations required for convergence of the optimization problem is less in the case of Multi-objective GA. The behavior observed can be utilized in reducing the computational time by lowering the number of required generations of the algorithm. When average computational times of fixed generations are considered, it is observed that both the Penalty-function GA and the Multi-objective GA perform almost equally well.

A modular approach has been followed in the implementation. The communication between the GA module and the neutronics simulation codes is achieved by the interface module in the middle. The GA module and the interface modules are developed in ‘C’ programming language. The interface module facilitates a smooth communication between the GA module and the neutronics simulation codes which are developed in FORTRAN programming language. The changes in the GA modules can easily be done without affecting the interface module. The modular approach has given the advantage of extending the implementation easily to the other similar burnup optimization studies.

CHAPTER 5

APPLICATION OF GENETIC ALGORITHM METHODOLOGIES IN CORE CONFIGURATION OPTIMIZATION STUDIES OF FAST BREEDER REACTORS

The multi objective, nuclear fuel management problems considered in this chapter aim to find out the optimal number of subassemblies in the reactor core, in order to achieve the best performance of the reactor. Two separate studies are conducted based on the cores of two different fast breeder reactors. Commensurate with the methodologies of Chapter 4, the Penalty-function GA and the Multi-objective GA are applied and compared.

5.1. INTRODUCTION

Fast breeder reactors play an important role in the three-stage nuclear power programme of India. Fast breeder reactors along with associated fuel reprocessing technologies, can help to ensure energy security in India. The milestone in the second stage is the 500 MWe Prototype Fast Breeder Reactor (PFBR), designed by Indira Gandhi Centre for Atomic Research (IGCAR) and is being commissioned at Kalpakkam, India [140, 141]. India is planning to construct more number of fast breeder reactors with improved economy and enhanced safety. In this direction, studies are being conducted at IGCAR towards the design of 500/600 MWe and 1000 MWe fast breeder reactors [142]. The configuration of the reactor core is an important aspect which has to be optimized for efficient, economic and safe production of power.

Finding out optimal core configuration of Fast Breeder Reactor (FBR), is the result of detailed neutronics scoping studies, taking into consideration of several factors like, size of the core, enrichment of the fuel, linear heat rating of the fuel pins, excess reactivity of the core, control rod design, and the inventory of the fuel. The optimization problem has multiple objectives and constraints, some of which are in conflict with each other. Any final solution inevitably requires some sort of compromise, in meeting the given objectives. Therefore, optimization of the core configuration design is a complex task in terms of computational effort and time.

The studies presented in this chapter are based on the cores of two different fast breeder reactors. The optimization studies are for the core configuration of:

(i) 500 MWe mixed-oxide fuelled reactor with two fuel enrichment zones

(ii) 1000 MWe metal-fuelled reactor with three fuel enrichment zones

Division of the core into different fuel enrichment zones is usually carried out during the design stage of the reactor, in order to achieve better power output [15]. The power output of the reactor can be increased by a more uniform distribution of the power density across the core. This is referred to as “flux flattening” which improves the average power and burnup of the peripheral subassemblies. Generally, the central zone of the core - where the neutron flux is highest - is loaded with fuel subassemblies of lower enrichment than that utilized in the outer zones. This may also provide improved utilization of the fuel and leading to decrease the cost of the power produced by the reactor.

The aim of the optimization problems of the 500 MWe and the 1000 MWe cores, considered in the present study, is to arrive at the optimal number of fuel subassemblies in different fuel enrichment zones of the reactor core, which gives the maximum fuel economy, while satisfying the operational and safety related constraints. The parameters which have significant influence in determining the optimal number of subassemblies of the core are selected as the objectives. These parameters are:

(i) excess reactivity of the core

(ii) linear heat rating of different fuel enrichment zones of the core

(iii) mass of the fuel required by considering the fuel enrichment of different zones

(iv) breeding ratio of the core

The excess reactivity of the core indicates the effective neutron multiplication factor to be provided in the core in order to override all the reactivity losses during an operational cycle. In the case of fast breeder reactor core, it includes the losses of reactivity due to temperature and power rise, core burnup and operating margin provided due to restrictions in the control rod movement [143]. The linear heat rating is the power generated per unit length of the fuel pin. The objective is to limit its value such that the temperature in the fuel pin does not exceed the melting point of the fuel. The fuel inventory represents the amount of fissile material used in the core and the objective is to get a core configuration with minimum fuel inventory, which has an impact on the fuel economy. The breeding ratio indicates the ratio of fissile material obtained to the fissile material spent. Higher breeding ratio is desirable to generate enough fissile material for self-sufficient closed fuel cycle of Fast Breeder Reactor programme.

The GA methodologies introduced in Chapter 2, i.e. Penalty-function GA and Multi-objective GA, are applied and studied in solve the optimization problems of the 500 MWe and the 1000 MWe cores. The optimization procedure followed in the two studies share several common features. For example, overall schemes of calculation as well as the neutronics simulation codes employed are the same for the two studies.

Therefore, the overall schemes of calculation of both the optimization problems can be considered together and are described in the following section.

5.2. OVERALL SCHEME OF CALCULATION FOLLOWED IN THE OPTIMIZATION PROCEDURE

The first step in applying GA to nuclear fuel optimization is to determine the representation method which is suitable for the problem. As part of GA representation, a candidate solution (in the present studies, the number of fuel subassemblies in different enrichment zones) is encoded as a digital chromosome which has enough information to reproduce the original solution. While being executed, GA generates a collection of trial solutions i.e. a population of chromosomes, and the fitness values of each chromosome is evaluated. For example, in the study of 500 MWe core, two integer numbers that represent number of subassemblies in the corresponding fuel enrichment zones of the core form one chromosome. Similarly, in the case of 1000 MWe core, three integer numbers that represent number of subassemblies in the three fuel enrichment zones of the core form one chromosome. The fitness value corresponding to each such chromosome is calculated by running the neutronics simulation codes. Subsequently, the fitness values calculated are used by the selection procedure of the GA.

The flowchart illustrating the overall scheme of calculation followed in the optimization procedure is given in Figure 5.1. The scheme of calculation includes GA module, interface module, and neutronics simulation codes. The optimization procedure is implemented with GA as the optimization module and the neutronics simulation codes, as the fitness evaluation module. The communication among these modules is smoothly achieved by the interface module. As shown in the flowchart, the interface module provides two-way communication between the GA module and the neutronics simulation codes.

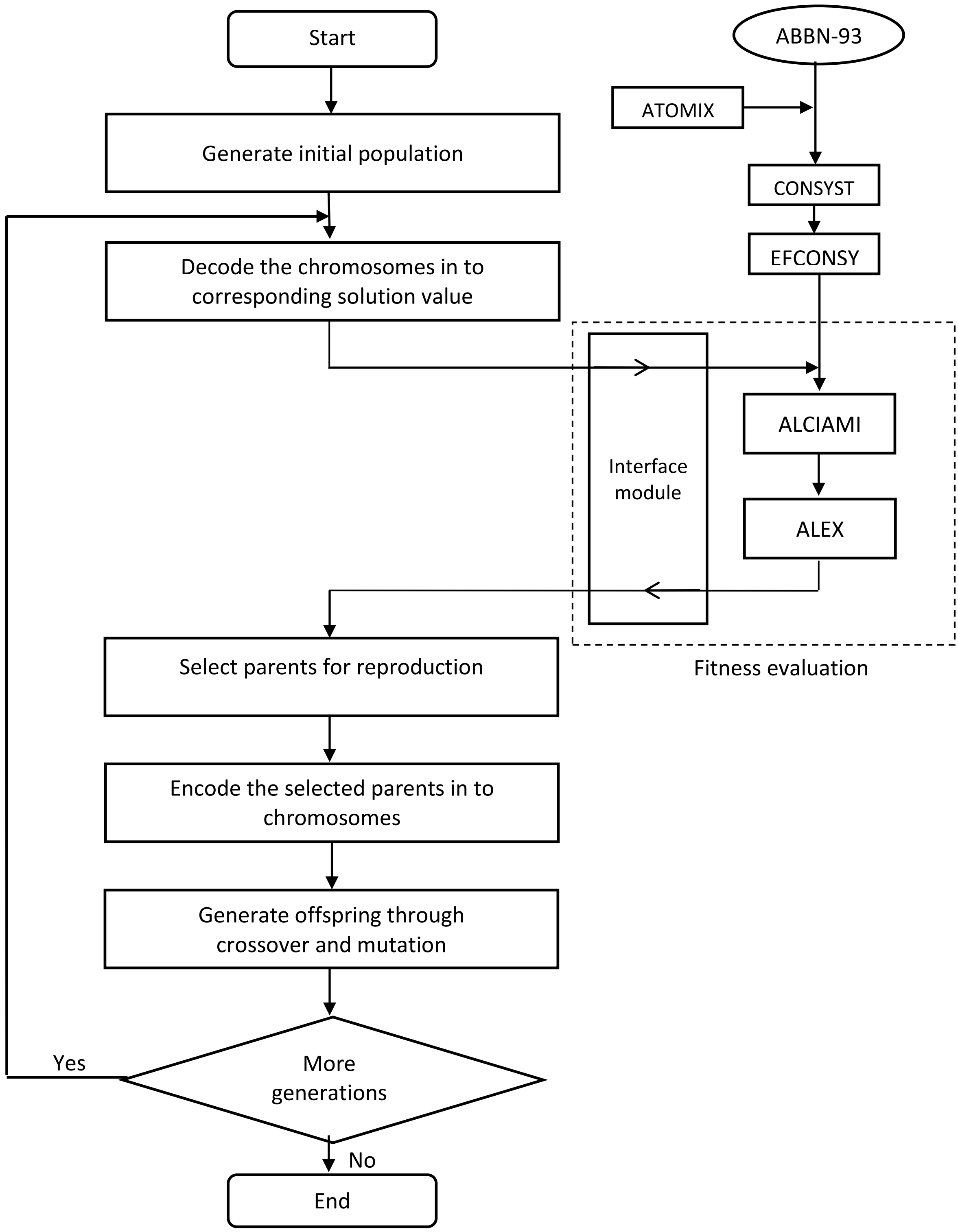


Figure 5.1: Flowchart of the overall scheme of calculation followed in the optimization procedure. ATOMIX, CONSYST, EFCONSY, ALCIALMI, and ALEX are the neutronics simulation codes. ABBN-93 is the multi group cross-section library.

The GA module is developed in 'C' programming language. Most of the neutronics codes used in the nuclear fuel management are in FORTRAN programming language and are specific to the type of reactor [51, 81]. The neutronics codes used in the present studies were also written in the FORTRAN programming language. The interface module should be compatible with the neutronics codes and also should be able to create the input files without user intervention. Similarly, the required output values generated by the neutronics simulation codes should be searched and read by the interface module and given back to the GA module for further calculations. The 'R' programming language [144, 145], which supports several efficient pattern searching and file-handling operators, is used in developing the interface module. Further, the 'R' programming language supports calling functions of the GA module during the runtime, as Dynamic Link Libraries (DLLs), which facilitates the smooth communication between the modules. Even though, a similar approach has been followed in the burnup optimization study presented in Chapter 4, the modular approach has been enhanced in the present study, by the use of the 'R' programming language. The enhanced modular approach gives novelty to the work carried out as part of the present studies, with certain advantages like:

(i) The GA modules as well as the neutronics simulation codes can easily be invoked from the interface module, during the runtime.

(ii) Interface modules having efficient pattern searching capabilities, can easily be modified to meet the requirement of different neutronics simulation codes. This allows extending optimization procedure to other studies of similar nature.

As has been seen during the discussion about the overall procedure of GA in nuclear fuel management (see Section 2.6 of Chapter 2), the neutronics simulation codes play a key role in predicting the reactor core behavior during the optimization procedure. The understanding about the basic functionalities and the overall working of neutronics simulation codes are achieved by conducting several independent trial runs of the codes. As depicted in Figure 5.1., there are five neutronics simulation codes used in the present studies which include ATOMIX, CONSYST, EFCONSY, ALCIALMI, and ALEX [143]. The number densities of various nuclei present in the different regions of the core are calculated using the code ATOMIX. Using the multi-group library of ABBN-93, self-shielded cross-sections are calculated by running CONSYST and EFCONSY codes [146, 147]. The excess reactivity of the core is calculated using the two-dimensional diffusion theory code, ALCIALMI, which uses R-Z geometry of the core for calculations. The code ALEX gives the power densities, from which linear heat rating of the fuel pins are calculated. The code ALEX also calculates breeding ratio of the given core configuration. In the present studies, the aim is confined to finding out the optimal number of fuel subassemblies, without varying the fuel enrichments of different zones. In every iteration of the optimization procedure, fuel inventory requirements are calculated based on the number of subassemblies assigned to the enrichment zones. Since the fuel enrichments of different zones are fixed in the present studies, the number density and cross-section calculations need not be repeated for every iteration of the fitness evaluation. Therefore, the codes - ATOMIX, CONSYST, EFCONSY - which are used for the number density and cross-section calculations, are represented outside the “Fitness evaluation” block in the flowchart (Figure 5.1). Even though the neutronics simulation codes employed are the same for optimizations of the 500 MWe core and the 1000 MWe core, these two studies differ in the following aspects:

(i) The types of fuel used is different, i.e. oxide fuel (UO_2-PuO_2) used in the 500 MWe core and metallic fuel (U-Pu-Zr) used in the 1000 MWe core. Therefore, the neutronics characteristics of the cores are different and hence, objectives and constraints defining the neutronics parameters of the studies are also different.

(ii) The power generation capacities of the cores are different. Therefore, the thermal characteristics of the cores are different and hence the objectives and constraints defining the thermal parameters of the studies are also different.

(iii) The number of fuel enrichment zones and the enrichment of fuel in the zones are different. Therefore, the models of the core employed in the optimization studies are different.

The overall scheme of calculation and the important differences between the optimization studies of 500MWe and 1000 MWe cores have been explained above. Next, the optimization of the 500 MWe core configuration is explained in detail.

5.3. OPTIMIZATION OF 500 MWe CORE CONFIGURATION

As mentioned earlier, the fast breeder reactor core of 500 MWe with two radial fuel enrichment zones is considered for the study. The model of the core used in the study is similar to the core of 500 MWe Prototype Fast Breeder Reactor (PFBR) in terms of:

(i) size and capacity of power generation of the core

(ii) types of fuel used

(iii) number of fuel enrichment zones present

(iv) types of subassemblies used

(v) neutronics and thermal characteristics of the core

The core configuration of the PFBR is already designed by conducting several detailed neutronics scoping studies using the conventional optimization methods (i.e., without employing intelligent optimization techniques like GA). The neutronics scoping studies were conducted by employing several neutronics simulation codes in order to predict the characteristics of the core. The same neutronics simulation codes are used in the present GA based optimization study (see Section 5.2 for the description about the codes). The inputs for the codes which define the required characteristics of the core are also the same. Therefore, the neutronics parameter arrived at for the PFBR core is used as a reference for verifying the results obtained from the present study. A brief description about the core of PFBR is given in the following section.

5.3.1. Reference Core Used for the Study : The Core of PFBR

The cross sectional view of the core of PFBR is shown in Figure 5.2 [143]. The core is composed of several types of subassemblies like fuel, control, blanket, and shielding. The fuel subassembly contains the mixed-oxide fuel (UO_2-PuO_2) with axial blanket and shield. The active core, where most of the nuclear heat is generated, consists of 181 fuel subassemblies. The active core (i.e. the fuel region) is divided in to two radial fissile enrichment zones: inner zone (referred to as core-1 in the rest of the study) and outer zone (referred to as core-2 in the rest of the study). The core-1 consists of 85 subassemblies with 21% PuO_2. Core-1 also houses 9 Control and Safety Rods (CSR) and 3 Diverse Safety Rods (DSR) for reactivity control and reactor shutdown. The core-1 is surrounded by 96 subassemblies of core-2, with relatively higher enrichment of 28% PuO_2. The variation of enrichments in the radial direction helps in radial flux flattening. In the axial direction, the fuel subassemblies mainly comprise of fuel material, upper axial blanket and lower axial blanket. The blanket subassemblies contain depleted uranium and the breeding happens in these subassemblies. The steel reflectors (denoted as ‘Steel SA’ in the figure) minimize leakage of neutrons from the core. The B_4C subassemblies shown in the figure are the neutron shielding subassemblies.

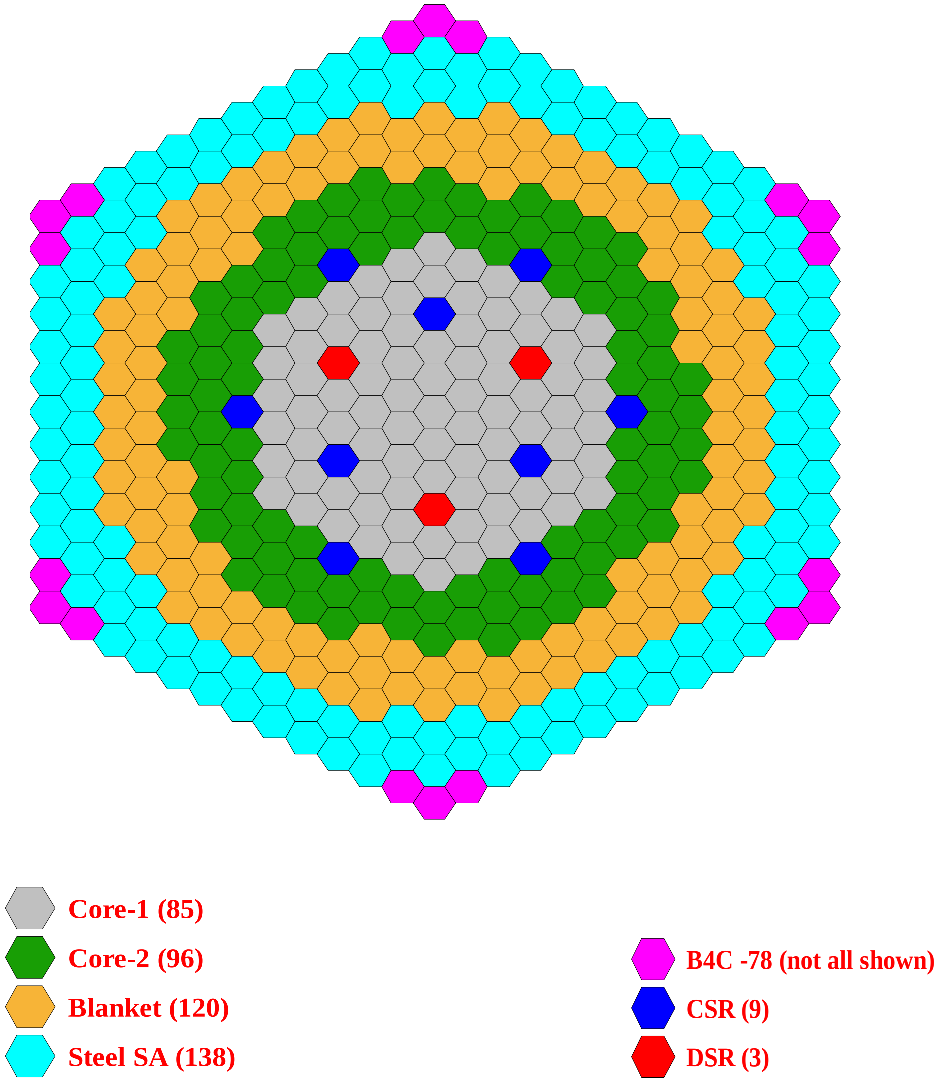


Figure 5.2: Cross sectional view of 500 MWe core of PFBR

There are certain differences between the core of PFBR and the model of the core employed in the present optimization procedure. The detail about the model of the core used in the optimization procedure is described in the following section.

5.3.2. Model of the Core Used in the Optimization Procedure

The neutronics simulation codes employed in the optimization procedure use two-dimensional core geometries which are based on R-Z models of the cores. The R-Z model of the 500 MWe core used for the study is shown in Figure 5.3.

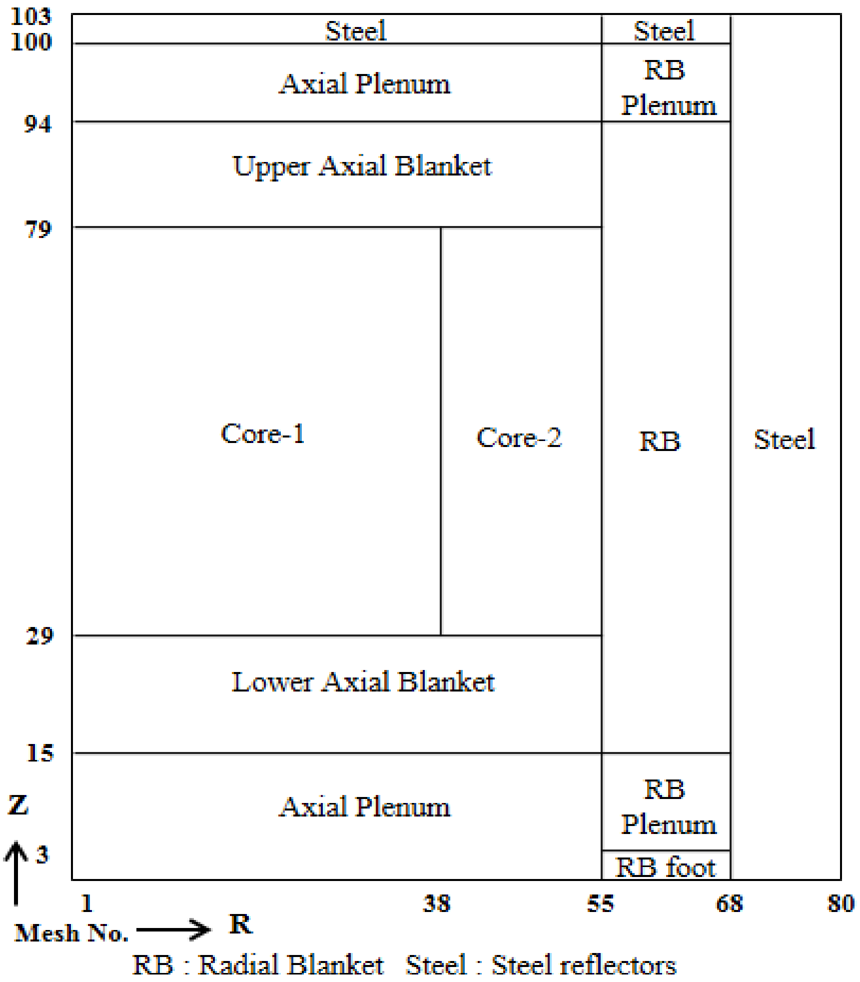


Figure 5.3. R-Z model of 500 MWe fast breeder reactor core used for the study (control rods, i.e. CSR and DSR, are not considered in the model)

The model shown in the figure differs from the core of PFBR (shown in Figure 5.2) by way of not considering the Control and Safety Rods (CSR) and Diverse Safety Rods (DSR). The presence of CSR and DSR is not having any significant influence on the objectives and the final solutions of the optimization problem. Further, the exclusion of CSR and DSR from the model for optimization allows varying the number of subassemblies in different enrichment zones of the core in an easier way. For finding out the optimal core configuration, the number of subassemblies placed in the enrichment zones are being changed in every iteration of the optimization procedure. Then, the evaluation of each of the configuration is carried out based on the objectives and constraints of the optimization problem. The above steps are to be repeated in the procedure without any manual intervention. The presence of CSR and DSR pose difficulty in achieving automatic variations in the model, and hence not considered in the optimization model.

As part of the optimization procedure, when the numbers of subassemblies in the enrichment zones are changed, the diameter of the core in the radial direction (i.e. radial width of the core) changes accordingly, while the height of the core remains unchanged. The variations in the radial width of the core during the optimization procedure are explained further in the following discussion. During the optimization procedure, when the diameter of the core changes, the radial width of the portions above and below the core-1 and core-2 (consisting of axial blankets, axial plenum and stainless steel reflector, see Figure 5.3) also vary accordingly. However, the radial width of the portions of the core beyond core-2 (consisting of radial blanket, radial blanket plenum, radial blanket foot and stainless steel reflector) remains unchanged. The radial width variation of the core geometry is more clearly depicted by the schematic representations in Figure 5.4.

The Figures 5.4(a) and 5.4(b) represent the radial widths of different core regions at two randomly selected iterations of the optimization procedure. In the figures, '$R1$' and '$R2$' denote the radii of core-1 and core-2 respectively. The terms '$C1$' and '$C2$' represent the radial width of blanket and steel reflectors respectively. The total radius of the core is denoted by '$R3$' i.e., $R3=R1+(R2-R1)+C1+C2$. During different iterations of the optimization procedure, '$R1$' and '$R2$' (hence $R3$ also) are varied independently but '$C1$' and '$C2$' are kept constant. Therefore, the optimization procedure that finds the optimal number of fuel subassemblies in core-1 and core-2 would represent the corresponding geometry of the total core also.

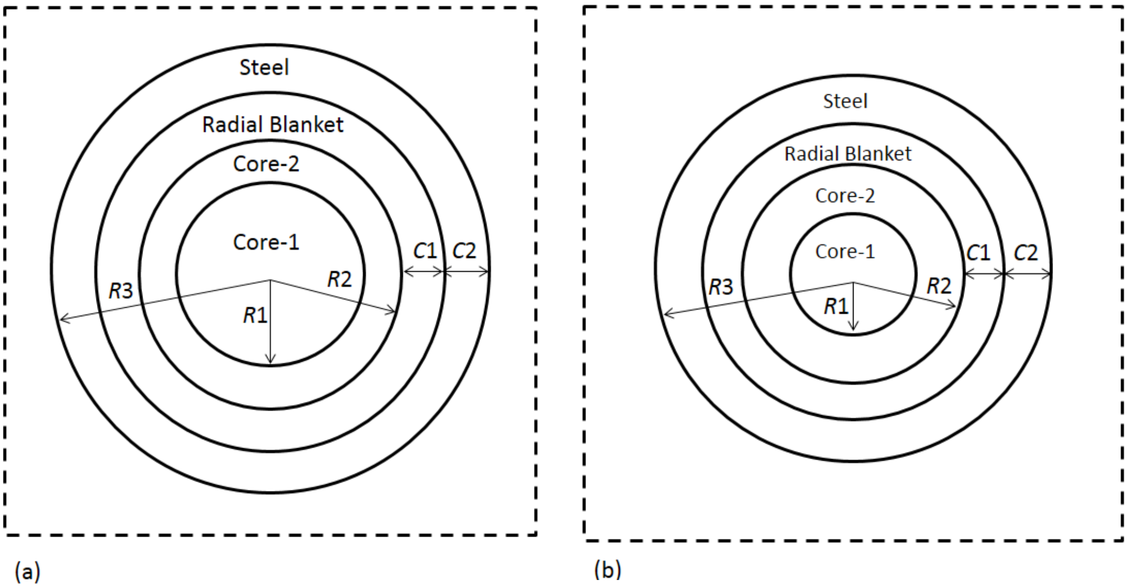


Figure 5.4: The schematic representations showing the radial width variations of the core regions for two different iterations in the optimization procedure. (a) Represents the iteration for bigger core geometry with higher values of *R*1, *R*2 and *R*3. (b) Represents the iteration for smaller core geometry with lower values of *R*1, *R*2 and *R*3.

Next, the details about mathematical model formulation of the optimization problem are taken up for consideration. The formulation of the generic mathematical model is considered first and then, the specific models for the Penalty-function GA and for the Multi-objective GA are considered. The models formulated are incorporated to the GA module of the optimization procedure.

5.3.3. Mathematical Model Formulation of the Optimization Problem

As already mentioned, the aim of the present study is to find the optimal number of subassemblies in core-1 and core-2 of a 500 MWe fast breeder reactor core. The optimal core configuration design is arrived at, while trying to satisfy the given objectives and constraints. The given optimization problem has five objectives and five constraints. The objectives for maximization are related to core excess reactivity (denoted by *RHO*) and breeding ratio (denoted by *BR*). The objectives for minimization are linear heat rating of core-1 (denoted by *LHR*1), linear heat rating of core-2 (denoted by *LHR*2), and percentage deviation of fuel inventory from a selected upper limit value (denoted by *FUI*). The objectives for minimization of linear heat ratings (i.e. *LHR*1 and *LHR*2) are to be within a specified limit for the selected problem. Generally, higher linear heat rating is better for minimizing the fuel inventory. However, the linear heat rating should not exceed a maximum upper limit by considering the structural integrity of the fuel pins. In the present study, the fresh core configuration at the initial startup of the reactor is considered. Subsequently, the minimization of linear heat ratings within the specified limits, in order to accommodate the effects of flux flattening happening during the prolonged operation of the reactor, is taken up. The unit of core excess reactivity is pcm (percent-milli) (1 pcm = $10^{-5}\frac{\Delta k}{k}$, where 'k'denotes the effective neutron multiplication factor, 'Δk' denotes its deviation from the unity) and that of linear heat rating is W/cm.

The upper and lower limits are defined for the constraints related to the parameters of *RHO*, *LHR*1, and *LHR*2. The constraint related to *FUI* has an upper limit and that of *BR* has a lower limit. The limits of the constraints are taken in accordance with the uncertainties involved in their estimation. A solution to the problem can be termed as feasible, only if it satisfies all the five constraints mentioned above. Accordingly, the mathematical formulation of the given optimization problem can be arrived as:

Max (*RHO*, *BR*) and *Min* (*LHR*1, *LHR*2, *FUI*)

= *f* (number of subassemblies of core-1,

number of subassemblies of core-2) (5.1)

Such that,

$$10800 \le RHO \le 11200 \text{ pcm}, \ 465 \le LHR1 \le 485 \text{ W/cm}, 430 \le LHR2 \le 460 \text{ W/cm},$$

$$FUI < \text{given upper limit (in \% deviation)}, \ BR > 1.045 \qquad (5.2)$$

where, *Max* represents the maximization, *Min* represents the minimization and $f()$ represents "function of". The given objectives are function of the number of subassemblies of core-1 and core-2. The number of subassemblies explored for the core-1 and the core-2 are fixed to certain ranges, arrived based on the results of initial trial runs of the neutronics simulation codes. Accordingly, the given problem has the two boundary conditions for the input values, as given below:

$$40 \le \text{number of subassemblies of core-1} \le 95 \qquad (5.3)$$

$$50 \le \text{number of subassemblies of core-2} \le 108 \qquad (5.4)$$

The generic mathematical model formulated above need to be further refined, in order to apply to the specific GA methodologies. The model formulations for the Penalty-function GA and for the Multi-objective GA are covered in the next section.

5.3.4. Model Formulation for the Penalty-function GA

In the case of Penalty-function GA, the multi-objective problem of fuel management optimization is converted in to a single objective by adding penalty functions and constraints. A detailed description about the Penalty-function GA is given in Chapter 2 (see Section 2.7.1). The model formulation of Penalty-function GA for the present optimization problem is considered here. Among the five objectives of the problem, the maximization of *BR* is taken as the primary objective for the Penalty-function GA and the other four objectives are converted to penalty functions. The penalized objective function for the selected problem is formulated as follows:

$$Fitness = BR - P1 - P2 - P3 - P4 \qquad (5.5)$$

where,

$P1 = |LHR1_{mid} - LHR1| \times A1,$ if $LHR1 < LHR1_{lb}$ or $LHR1 > LHR1_{ub}$

$= [LHR1 - (LHR1_{ub} + 1)] \times A1,$ otherwise

$P2 = |LHR2_{mid} - LHR2| \times A2,$ if $LHR2 < LHR2_{lb}$ or $LHR2 > LHR2_{ub}$

$= [LHR2 - (LHR2_{ub} + 1)] \times A2,$ otherwise

$P3 = |RHO_{mid} - RHO| \times A3,$ if $RHO < RHO_{lb}$ or $RHO > RHO_{ub}$

$= [(RHO_{ub} + 1) - RHO] \times A3,$ otherwise

$P4 = (FUI - FUI_{ub}) \times A4$

*P*1 : penalty function related to *LHR*1

*P*2 : penalty function related to *LHR*2

*P*3 : penalty function related to *RHO*

*P*4 : penalty function related to *FUI*

Fitness: penalized objective function used for the fitness evaluation in GA

*A*1, *A*2, *A*3, *A*4: constant values selected to give proper weightage to the corresponding penalty functions

In the above equations, the subscripts have the following meanings related to the corresponding objective functions:

mid : middle value of the feasible range

lb : lower bound value of the feasible range

ub : upper bound value of the feasible range.

The penalty functions (*P*1, *P*2, *P*3 and *P*4*)* are formulated in such a way that, if the objectives fall with in the corresponding feasible range, then a positive value is added to the *BR* to get a higher "*Fitness*" value. On the other hand, if the objectives fall above or below the feasible range, then a negative value is added to the *BR* to get a lower "*Fitness*" value.

5.3.5. Model Formulation for the Multi-objective GA

Based on the generic mathematical model, the multi-objective GA has been implemented (NSGA-II implementation) to find the optimal number of subassemblies in core-1 and core-2. A detailed description about the procedure of NSGA-II is covered in Chapter 2 (see Section 2.7.4). One important step in the multi-objective GA is the handling of constraint violations, which helps the algorithm to bias the search through a constrained space. The constraint violations are handled by an approach which is similar to the penalty handling mechanism in the Penalty-function GA. The constraint functions are first normalized and then the violation for each constraint is calculated. For the five constraints of the selected problem, corresponding constraint violations are calculated as:

$$
\begin{aligned}
C1 &= \frac{RHO - RHO_{mid}}{RHO_{min} - RHO_{mid}}, && \text{if } RHO < RHO_{lb} \\
&= \frac{RHO - RHO_{mid}}{RHO_{max} - RHO_{mid}}, && \text{if } RHO > RHO_{ub} \\
&= 0, && \text{otherwise}
\end{aligned} \tag{5.6}
$$

$$
\begin{aligned}
C2 &= \frac{LHR1 - LHR1_{mid}}{LHR1_{min} - LHR1_{mid}}, && \text{if } LHR1 < LHR1_{lb} \\
&= \frac{LHR1 - LHR1_{mid}}{LHR1_{max} - LHR1_{mid}}, && \text{if } LHR1 > LHR1_{ub} \\
&= 0, && \text{otherwise}
\end{aligned} \tag{5.7}
$$

$$
\begin{aligned}
C3 &= \frac{LHR2 - LHR2_{mid}}{LHR2_{min} - LHR2_{mid}}, && \text{if } LHR2 < LHR2_{lb} \\
&= \frac{LHR2 - LHR2_{mid}}{LHR2_{max} - LHR2_{mid}}, && \text{if } LHR2 > LHR2_{ub} \\
&= 0, && \text{otherwise}
\end{aligned} \tag{5.8}
$$

$$
\begin{aligned}
C4 &= \frac{FUI - FUI_{u}}{FUI_{max} - FUI_{u}}, && \text{if } FUI > FUI_{u} \\
&= 0, && \text{otherwise}
\end{aligned} \tag{5.9}
$$

$$C5 = \frac{BR_l - BR}{BR_l - BR_{min}}, \quad \text{if } BR < BR_l$$
$$= 0, \quad \text{otherwise} \tag{5.10}$$

where, the terms $C1$, $C2$, $C3$, $C4$ and $C5$ represents the constraint violation values related to *RHO*, *LHR*1, *LHR*2, *FUI* and *BR* respectively. In the above equations, the subscripts have the following meanings related to the corresponding objective functions:

min : minimum value possible

max : minimum value possible

mid : middle value of the feasible range

lb : lower bound value of the feasible range

ub : upper bound value of the feasible range

l : lower limit value

u : upper limit value.

According to the above procedure, the individual constraint violations corresponding to the five constraints of the problem are calculated. Then, the overall constraint violation (C_{tot}) is calculated as:

$$C_{tot} = C1+C2+C3+C4+C5 \tag{5.11}$$

The next step is to modify each objective function value, according to the overall constraint violation. The overall constraint violation is multiplied with suitable constant values and the product is added to each of the objectives, to get the modified values of the objectives as:

$$RHO_{mod} = RHO + B1 \times C_{tot} \tag{5.12}$$

$$LHR1_{mod} = LHR1 + B2 \times C_{tot} \tag{5.13}$$

$$LHR2_{mod} = LHR2 + B3 \times C_{tot} \tag{5.14}$$

$$FUI_{mod} = FUI + B4 \times C_{tot} \tag{5.15}$$

$$BR_{mod} = BR + B5 \times C_{tot} \tag{5.16}$$

where, the term C_{tot} represents the overall constraint violation. The subscript '*mod*' denotes the modified values of the objectives. The terms *B*1, *B*2, *B*3, *B*4, and *B*5 are used to denote constant values, assigned to make both terms on the right side of the above equations to have the same order of magnitude. For a feasible solution, C_{tot} should be 0 and in that case, the modified values of the objectives are same as that of actual objective function values. For an infeasible solution, a penalty is added to each of the objective functions, corresponding to their constraint violations. Once the modified objective functions are calculated, those values are used by the multi-objective GA for Pareto-optimal fronts sorting.

The details of formulations of the optimization models suitable for the two GA methodologies are discussed above. The next step is to implement the models by following the overall scheme of calculation of the optimization procedure. The details of implementation of the optimization models are considered in the following section.

5.3.6. Details of the Implementations

As part of the study, a comparison is made by considering the performance of the Penalty-function GA and the Multi-objective GA. Therefore, the representation schemes and parameter values of GA, selected for the implementation of the algorithms are kept the same. The parameters, those are kept the same for both the algorithms, are:

(i) encoding scheme of GA

(ii) population size

(iii) crossover mutation methods and their probability

(iv) maximum number of generations

The values assigned for these parameters are given in Table 5.1. The real-number representation is selected for the implementation of both the Penalty-function GA and the Multi-objective GA. In the study, the decision variables (i.e. number of subassemblies in the different enrichment zones of the core) are not real numbers, but are integers. Therefore, the truncated values of the real numbers are encoded in the chromosomes. The study has been carried out on a computer system with Intel Core2 Duo CPU@3GHz and 2 GB RAM. The implementation of algorithms (both for the Penalty Function-GA and for the Multi-objective GA) are carried out by using the 'C' programming language and the interface module is implemented using the 'R' programming language. The initial population is generated by calling the random number generation function of the 'C' programming language. Several trial runs were conducted with randomly generated initial population of the GA, ensuring different initial search space. The results of implementations of the algorithms are discussed next.

Table 5.1: Genetic parameters and methods or values used in the study. The parameters given in the table are kept the same for Penalty-function GA and Multi-objective GA

Parameter	**Methods/Values**
Encoding	Floating point
Population size	40
Crossover method	Arithmetical
Crossover probability (CR)	0.6
Mutation method	Non-uniform
Mutation probability (MR)	0.025
Maximum number of generations	50

5.4. RESULTS OF THE STUDY OF 500 MWe CORE CONFIGURATION

Both of the GA methodologies, i.e. the Penalty Function-GA and the Multi-objective GA, are studied in detail to ascertain the suitability for the selected problem. Based on the results generated initially, GA parameters and penalty coefficients are fine-tuned. The fine-tuned versions of the algorithms are used to generate final results by conducting ten trial runs for each of the algorithms.

The core configuration of the Prototype Fast Breeder Reactor (PFBR) is considered as the reference for the present study. Therefore, the number of fuel subassemblies designed for the PFBR core is compared with the results arrived from the optimization study, and presented in Table 5.2. The average of the number of subassemblies arrived from the ten trial runs for each of the algorithms is furnished in the table. All of the feasible solutions arrived at the final generation (50^{th} generation) are considered and the average number of subassemblies in core-1 and core-2 are calculated. The values given in the table are arrived at, by rounding the average values (which are real numbers) to the nearest integers. It can be observed that both of the GA methodologies, i.e. the Penalty Function-GA and the Multi-objective GA, are able to generate feasible solutions which are agreeing with the number of subassemblies of the reference core.

Table 5.2: Comparison of the results obtained from the present study with the number of fuel subassemblies of the reference core (core of PFBR).

Fuel enrichment zones	Number of subassemblies		
	Reference core: PFBR	Results of the present study	
		Penalty-function GA	Multi-objective GA
Core-1	85	86	85
Core-2	96	95	95

The small variations in the results are attributed to the stochastic nature of the genetic algorithms and also to the approximations employed in the optimization models.

5.4.1. Convergence of Objective Functions : Final Generation

The feasible solutions obtained in the final generation (i.e. 50^{th} generation) of the Penalty Function-GA and the Multi-objective GA, generated by the ten trial runs of each of the algorithms, are given in Tables 5.3 and 5.4. The maximum and minimum values of each of the objectives, from the final generations of the algorithms, are presented in the tables. Corresponding number of subassemblies arrived, as the solutions to the optimization problem, are also given in the table. The outputs generated by neutronics simulation codes are represented with the accuracy of two decimal places for *RHO*, *LHR*1, and *LHR*2. In the case of *FUI*, the percentage deviation is calculated from the given upper limit and represented with the accuracy of two decimal places. The *BR* is represented with the accuracy of four decimal places, for proper representation of deviations in their values.

The consolidated results obtained, by considering the feasible solutions generated by the Penalty Function-GA and the Multi-objective GA, in the final generation of ten trial runs are furnished in Table 5.5. The maximum (Max), minimum (Min), average (Ave) and standard deviation (SD) values for the five objectives are given in the table. The maximum values are calculated by taking the average of maximum values produced (separately for each of the five objective functions) at the final generation for the ten trial runs. Similarly, the minimum values are calculated by taking the average of minimum values generated. The average values shown in the table are calculated by finding the average values of respective objectives (by considering the 50 members of the final population) for the ten trial runs and then, the corresponding standard deviations are calculated.

Table 5.3: Maximum and minimum values obtained in the final generation for the five objective functions and the corresponding solutions (Penalty-function GA)

Trial No.	Max /Min	Values obtained for objective functions					Solutions arrived	
		RHO	*LHR*1	*LHR*2	*FUI*	*BR*	Core-1	Core-2
1	Max	10860.15	477.58	441.14	-9.18	1.0611	90	94
	Min	10838.62	472.43	431.44	-7.82	1.0605	87	94
2	Max	11041.68	472.09	436.61	-8.18	1.0589	89	96
	Min	10951.69	466.71	432.18	-7.18	1.0567	88	95
3	Max	10987.33	482.90	451.19	-10.55	1.0617	85	95
	Min	10885.44	477.28	446.45	-9.55	1.0595	84	94
4	Max	11157.74	479.33	444.45	-9.64	1.0613	87	97
	Min	10868.13	466.39	437.26	-7.55	1.0551	86	94
5	Max	10943.84	474.13	434.64	-8.27	1.0607	90	95
	Min	10845.40	468.72	430.24	-7.27	1.0585	89	94
6	Max	11178.23	475.25	448.41	-9.45	1.0576	85	97
	Min	11078.31	469.78	443.73	-8.45	1.0554	84	96
7	Max	10943.84	474.13	434.64	-8.27	1.0607	90	95
	Min	10845.40	468.72	430.24	-7.27	1.0585	89	94
8	Max	11167.78	468.07	440.48	-8.00	1.0552	87	97
	Min	11157.74	466.39	437.26	-7.55	1.0551	86	97
9	Max	11088.59	480.85	453.19	-10.45	1.0598	85	96
	Min	10987.33	473.51	445.08	-9.00	1.0574	83	95
10	Max	11178.23	484.71	454.62	-11.00	1.0619	85	97
	Min	10885.44	469.78	443.73	-8.45	1.0554	83	94

Table 5.4: Maximum and minimum values obtained in the final generation for the five objective functions and the corresponding solutions (Multi-objective GA)

Trial No.	Max /Min	Values obtained for objective functions					Solutions arrived	
		RHO	*LHR*1	*LHR*2	*FUI*	*BR*	Core-1	Core-2
1	Max	11189.17	484.71	456.63	-11.00	1.0619	87	97
	Min	10860.15	471.51	439.85	-8.64	1.0555	82	94
2	Max	11110.48	482.90	456.63	-10.91	1.0617	88	96
	Min	10860.15	471.78	436.61	-8.18	1.0573	82	94
3	Max	11189.17	484.71	455.17	-11.00	1.0619	90	97
	Min	10838.62	468.07	430.24	-7.27	1.0552	82	94
4	Max	11189.17	484.71	456.63	-11.00	1.0619	90	97
	Min	10838.62	469.77	431.44	-7.73	1.0554	82	94
5	Max	10997.37	484.71	454.62	-11.00	1.0619	86	95
	Min	10868.13	477.28	444.45	-9.55	1.0595	83	94
6	Max	11189.17	484.71	458.60	-11.00	1.0619	89	97
	Min	10852.56	466.39	432.18	-7.18	1.0551	81	94
7	Max	11088.59	484.71	456.63	-11.00	1.0619	87	96
	Min	10868.13	473.51	439.85	-8.64	1.0574	82	94
8	Max	11178.23	484.71	458.60	-11.00	1.0619	89	97
	Min	10852.56	466.71	432.18	-7.18	1.0552	81	94
9	Max	11189.17	482.90	458.60	-11.00	1.0617	89	97
	Min	10852.56	469.78	433.41	-7.73	1.0554	81	94
10	Max	11189.17	484.71	456.63	-10.91	1.0619	85	97
	Min	10876.57	471.51	445.08	-8.91	1.0555	82	94

Table 5.5: Values of feasible solutions arrived at the final generation (for ten trial runs) of the Penalty-function GA and the Multi-objective GA

		Max	Min	Ave	SD
Penalty-function GA	*RHO*	11054.74	10934.35	10977.12	41.68
	*LHR*1	476.90	469.97	474.45	2.21
	*LHR*2	443.94	437.76	441.69	1.87
	FUI	-9.30	-8.01	-8.85	-99.6
	BR	1.0599	1.0572	1.0590	0.0009
Multi-objective GA	*RHO*	11150.97	10856.81	10990.25	95.52
	*LHR*1	484.35	470.63	477.26	4.23
	*LHR*2	456.87	436.53	446.95	5.99
	FUI	-10.98	-8.10	-9.57	-99.12
	BR	1.0619	1.0562	1.0593	0.0018

The convergence of feasible solutions at the final generation by the two algorithms can be clearly understood from the box plots presented in Figure 5.5. The results generated at the final generation of the Penalty-function GA and the Multi-objective GA by the ten trial runs are given in the box plots. The distributions of the five objectives for the Penalty-function GA are shown in Figure 5.5(a) and those for the Multi-objective GA are shown in Figure 5.5(b). The vertical height of the boxes represents the spread of feasible solutions for 50% of the data samples.

The major observations from the results presented in the Tables 5.3 to 5.5 and the Figure 5.5 are:

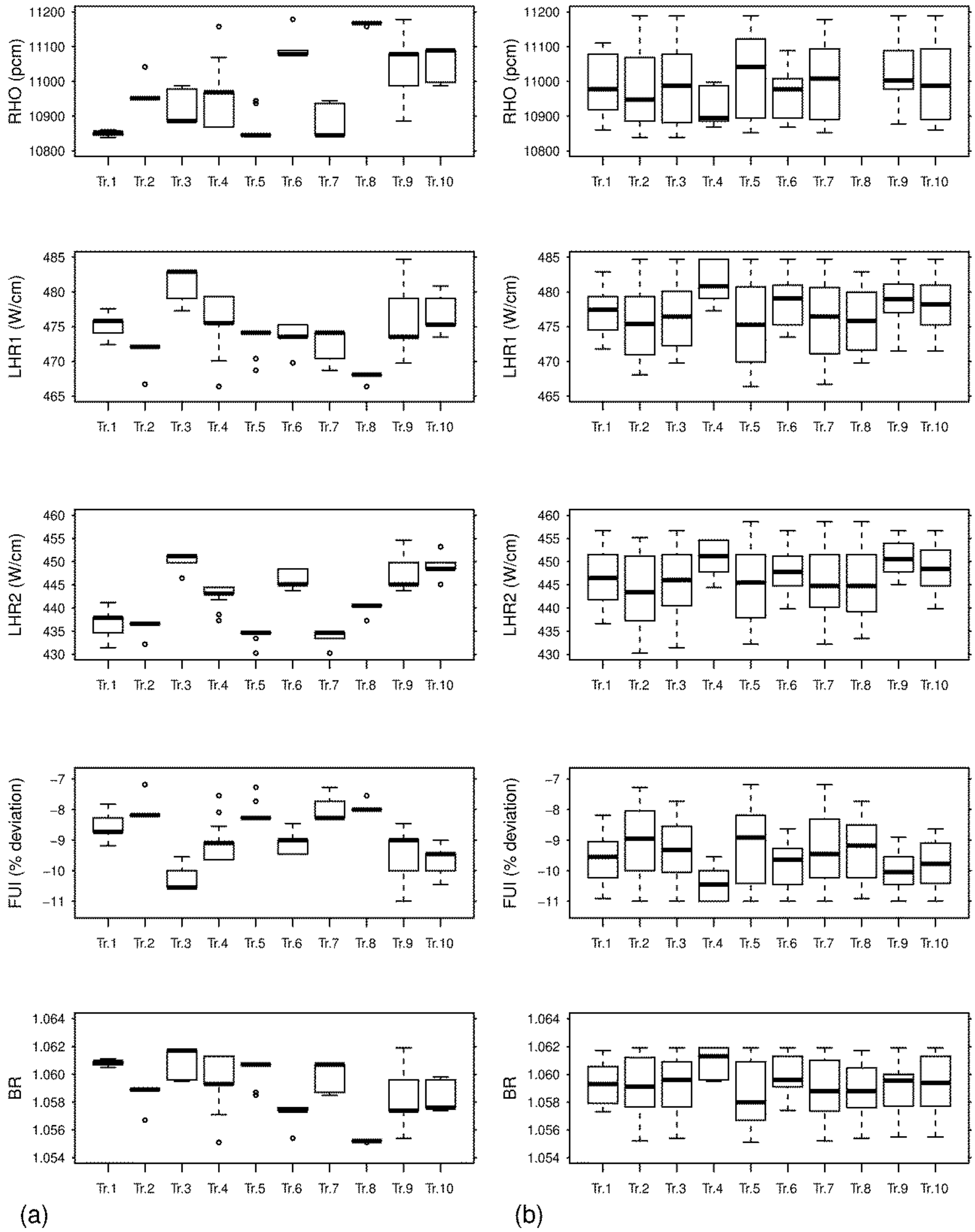


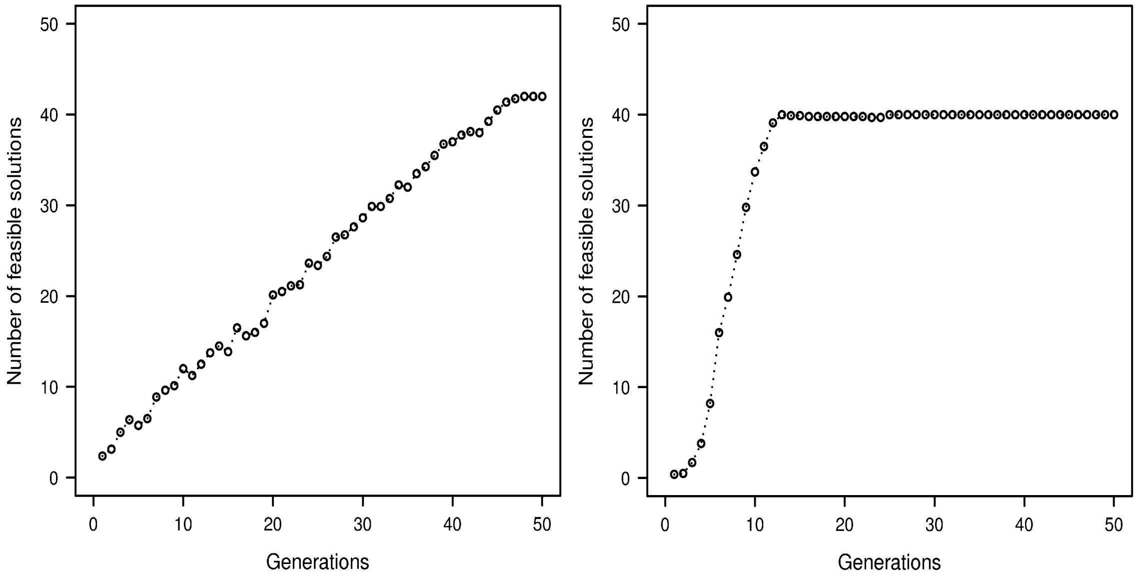
Figure 5.5: Box plots to represent the distribution of the five objectives among generated feasible solutions in the final generation of ten trial runs: (a) for Penalty-function GA. (b) for Multi-objective GA.

- Both of the algorithms, i.e. the Penalty Function-GA and the Multi-objective GA, are able to produce feasible solutions consistently at the final generation.
- The feasible solutions generated by the Penalty Function-GA lies in narrower range as compared with the Multi-objective GA.

5.4.2. Comparison of GA Performance: Generation wise Production of Feasible Solutions

A performance comparison has been done by considering the number of feasible or good solutions produced in successive generations of the Penalty-function GA and the Multi-objective GA. In the present optimization problem, one generated solution is termed as feasible, if it satisfies all the five constraints. That is, a feasible solution satisfies the conditions such as, *RHO* should be in the range between 10800 and 11200 pcm, *LHR*1 should be in the range between 465 and 485 W/cm, *LHR*2 should be in the range between 430 and 460 W/cm, *FUI* should be less than the given upper limit and *BR* should be greater than 1.045.

Figure 5.6 shows the average number of feasible solutions produced in successive generations by ten trial runs of the Penalty-function GA (Figure 5.6(a)) and ten trial runs of the Multi-objective GA (Figure 5.6(b)). The conditions of feasibility are kept the same for both the algorithms. It can be observed from the figures that the Multi-objective GA produces feasible solutions at a faster rate for earlier generations and has a better convergence speed. A comparison is made on GA performance based on the average CPU time taken to produce equal number of feasible solutions.



(a) (b)

Figure 5.6: Average Number of feasible solutions (generation wise) of the ten trial runs: (a) for Penalty-function GA. (b) for Multi-objective GA.

The average times and generations taken for producing 80% of population with feasible solutions (i.e. 32 feasible solutions out of the 40 members of the population) are taken in the comparison. The Penalty-function GA took 35 generations and 6564.60 seconds to produce 32 feasible solutions. The Multi-objective GA took 10 generations and 1807.90 seconds to produce the same number of 32 feasible solutions; implying that the Multi-objective GA is 73% faster than Penalty-function GA, with respect to CPU time for generating 80% of the population with feasible solutions.

The behavior observed can be utilized in reducing the computational time by lowering the number of required generations of the Multi-objective GA. Another observation is that, when the computational time requirement for a fixed number of generations are considered, there is no significant difference between the Penalty-function and the Multi-objective GA. The average CPU time requirement (for ten trial runs, each with 50 generations) for the Penalty-function GA is 9377.92 seconds whereas for the Multi-objective GA it is 9039.40 seconds. It is seen that, when average computational time of fixed number of generations are considered, the Multi-objective GA has a marginal advantage being 4% faster than the Penalty-function GA. The overall observation from the comparison is that the Multi-objective GA has a significant advantage in convergence speed and has a marginal advantage in computational time, when compared with the Penalty-function GA.

5.4.3. Generation wise Evolution of Objective Functions

The convergences of the five objectives during the whole evolution of the algorithms are presented in Figure 5.7. The solid black points denote the average value of the objective for the whole population of that particular generation. The dotted lines above and below denote the standard deviation. The convergences of the five objectives for the Penalty-function GA are shown in Figure 5.7(a) to 5.7(e). Similarly, the convergences of the five objectives of the Multi-objective GA are shown Figure 5.7(f) to 5.7(j). Following observations can be made by comparing the results:

- Both of the algorithms, i.e. Penalty-function GA and Multi-objective GA, are capable of arriving at around the same converging area of feasible solutions with respect to the given objectives.
- The objective functions are converging to a narrower region towards the end of generations for the Penalty-function GA as compared to the Multi-objective GA.

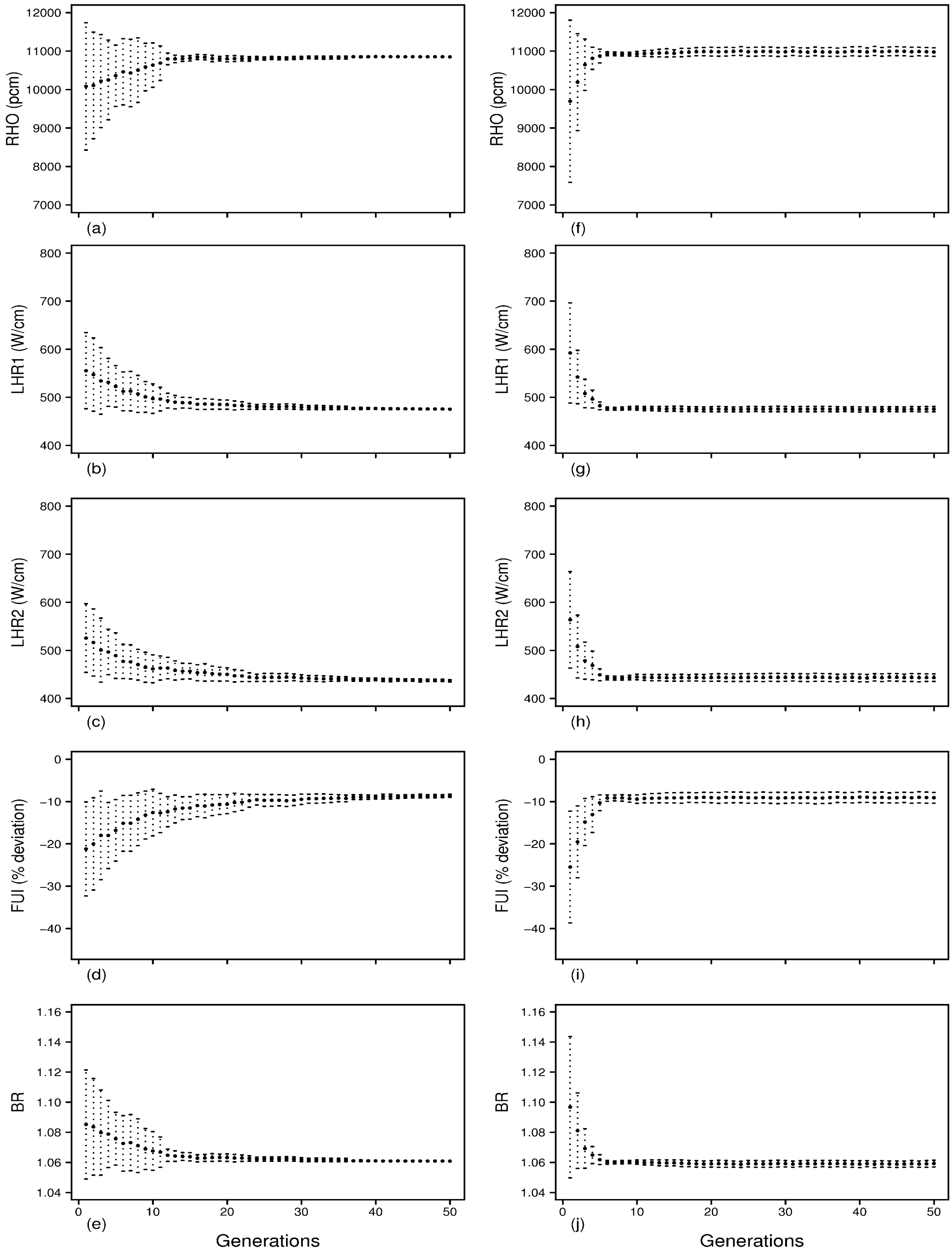


Figure 5.7: Convergences of the five objectives during the generation wise evolution of the algorithms: (a) to (e) for Penalty-function GA. (f) to (j) for Multi-objective GA.

- The speed of convergence is faster for the Multi-objective GA as compared to the Penalty-function GA.
- In the case of Multi-objective GA, for all the five objectives, the standard deviations (shown as dotted lines) are shrinking between the 5th and 10th generations and are getting wider in the later generations. This indicates the ability of the Multi-objective GA in converging faster to a feasible area in the search space and then, exploring more diversity around that area towards the end of the generations.

5.4.4. Solution Diversity: Initial Population Vs. Final Population

The number of subassemblies in core-1 and in core-2, forms the solution candidates for the optimization problem. The distributions of solution candidates, in the population of initial generation and also in the population of final generation, with respect to the overall search space are compared in Figure 5.8. The solution diversity of the initial and the final populations for the Penalty-function GA are shown in Figure 5.8(a) and that of the Multi-objective GA are shown in Figure 5.8(b). The total population and the corresponding solution candidates of a single trial run are considered here. The solution values converge to a narrow area in case of the Penalty-function GA, but to a wider area for the Multi-objective GA. This indicates, as in the case of diversity of the objectives, that the diversity of solutions is also better for the Multi-objective GA.

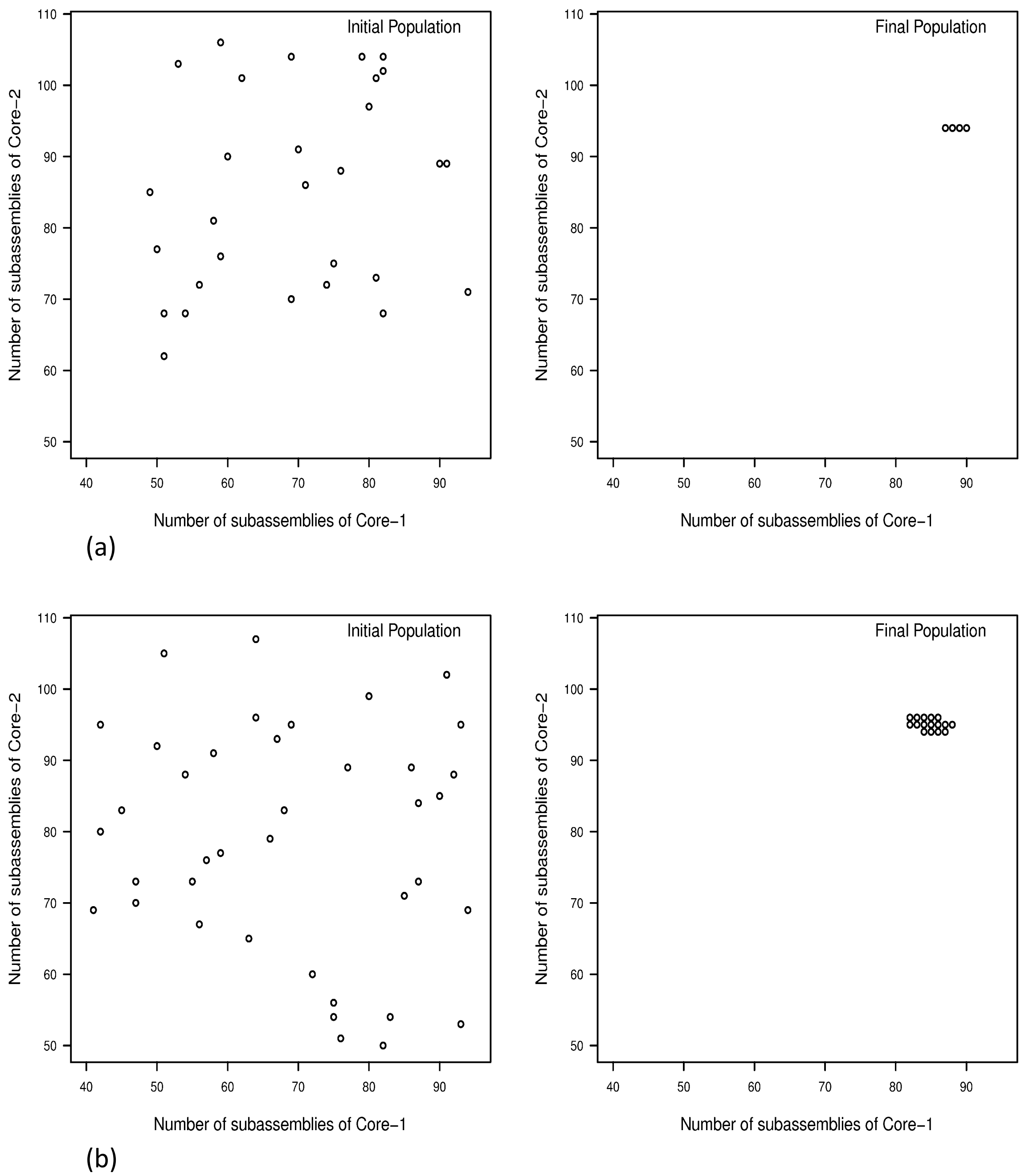


Figure 5.8: Comparison of solution diversity of the algorithms (initial population vs. final population) (a) for Penalty-function GA. (b) for Multi-objective GA.

The optimization study presented above deals with the 500 MWe fast breeder reactor core. Details of the study of 1000 MWe core are presented next.

5.5. OPTIMIZATION OF 1000 MWe CORE CONFIGURATION

As mentioned earlier, the 1000 MWe fast breeder reactor with metal-fuelled core is considered for the present GA based optimization study. In fast breeder reactors, the basic thermal and neutronics performance of metal fuels (U-Pu-Zr) is better than oxide fuel [148]. Therefore, several studies are being conducted at IGCAR towards the design of metal-fuelled fast breeder reactors [148, 149, 150]. The core configuration of one such reactors, proposed by Riyas and colleagues, is considered as the reference for the present optimization study [148] and a brief description of the core is given in the following section.

5.5.1. Reference Core Used for the Study

The proposed 1000 MWe fast breeder reactor core is having three radial fuel enrichment zones [148]. Generally, for better flux flattening and fuel utilization, the number of fuel enrichment zones increases as the core size increases [15]. Therefore, three fuel enrichment zones are envisaged for the 1000 MWe core, whereas only two enrichment zones were there for the 500 MWe core. The cross sectional view of the core representing the radial distribution of subassemblies is shown in Figure 5.9. The fuel subassemblies contains metallic ternary alloy of U-Pu-Zr with axial blanket and shield. The active core is divided in to three radial fissile enrichment zones viz., core-1, core-2 and core-3. The core-1 houses 79 subassemblies with 12% of fuel enrichment, core-2 has 96 subassemblies with 13% of enrichment and core-3 has 72 subassemblies with 18% of enrichment. There are 18 numbers of Control and Safety Rods (CSR) and 6 numbers of Diverse Safety Rods (DSR) placed in the core, as shown in Figure 5.9.

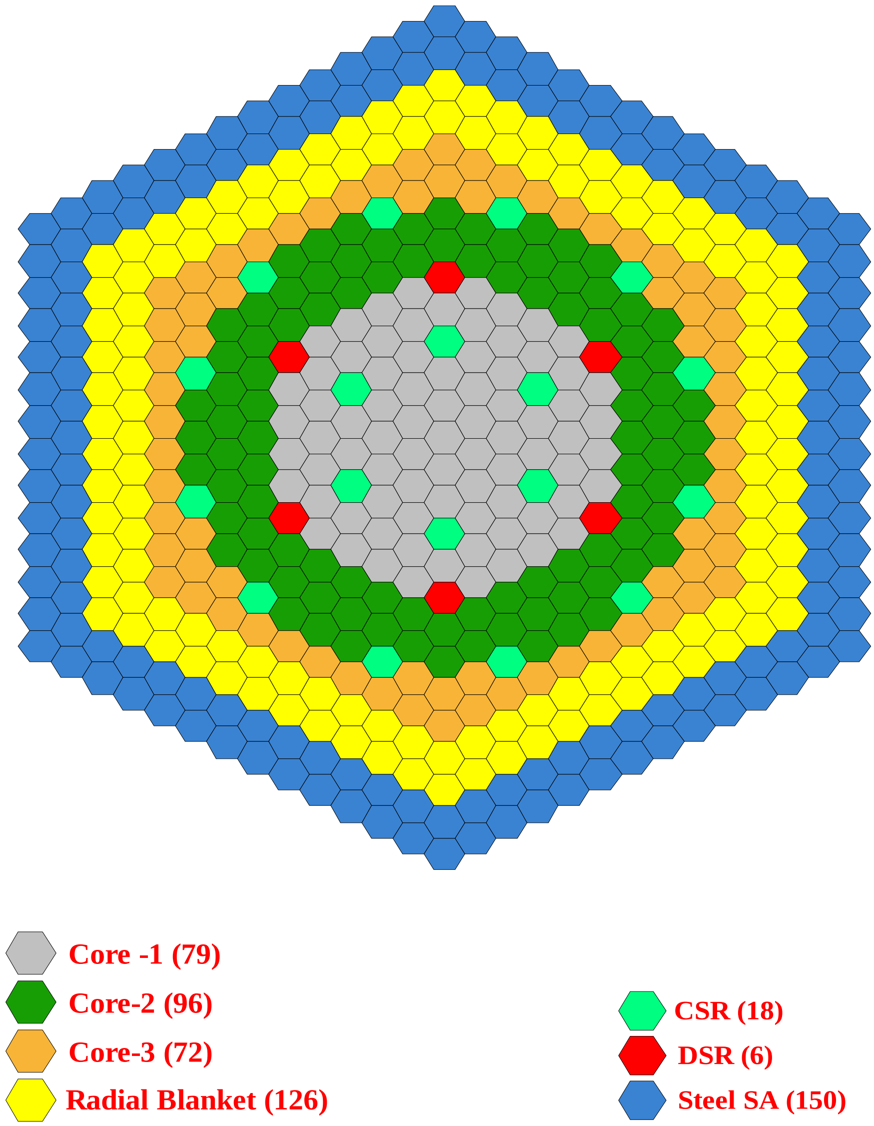


Figure 5.9: Cross sectional view of 1000 MWe Fast Breeder Reactor core

The fuel subassemblies comprise of fuel material, upper blanket and lower blanket, in the axial direction. In the radial direction, the peripheral region of the core contains redial blankets and steel reflectors (denoted as 'Steel SA' in the figure).

The model of the core, discussed above is similar to the model used for the present optimization study, but not exactly the same. The actual model of the core used in the study is discussed in the next section.

5.5.2. Model of the Core Used in the Optimization Procedure

The neutronics simulation codes employed in the optimization procedure use two-dimensional core geometries which are based on the R-Z model of the 1000 MWe core. The R-Z model of the core used for the study is shown in Figure 5.10. As in the case of the optimization study of the 500 MWe core, the model used in the present study differs from the reference core of 1000 MWe by way of not considering the control rods, i.e. Control and Safety Rods (CSR) and Diverse Safety Rods (DSR).

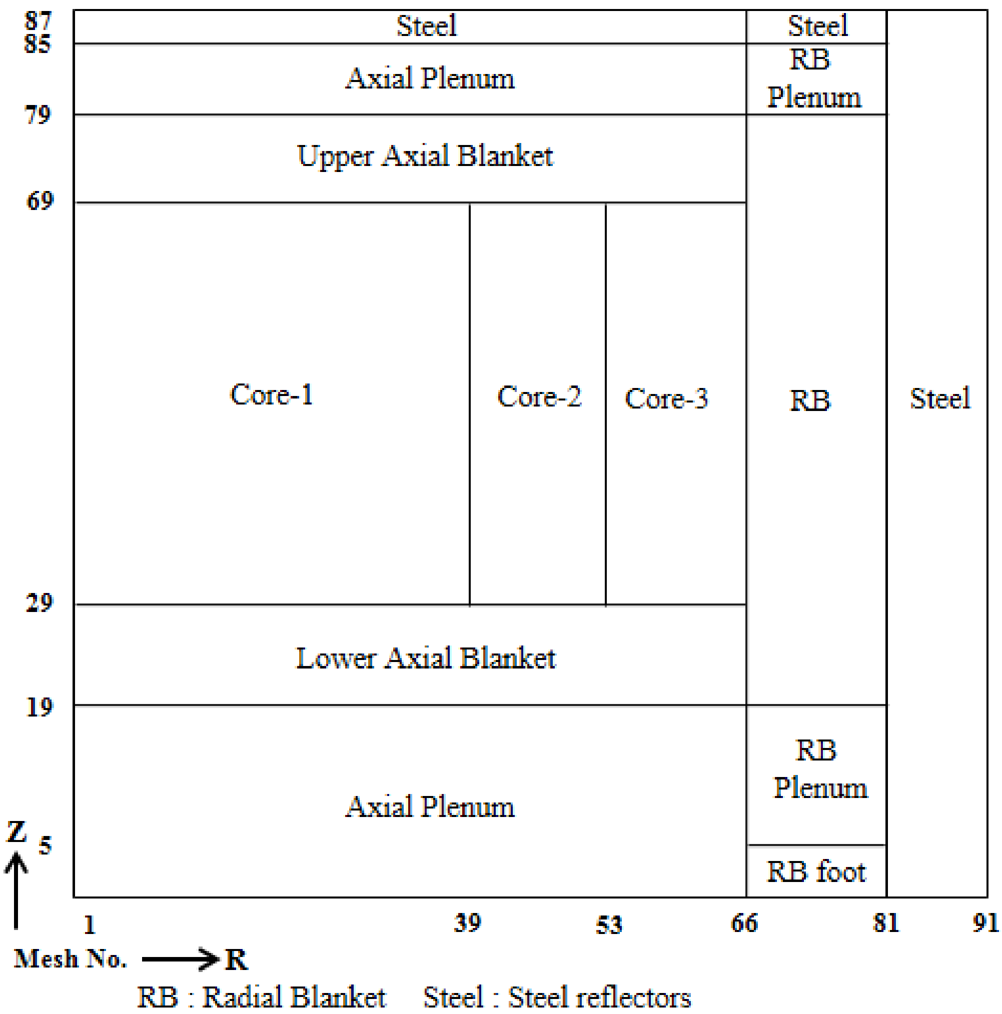


Figure 5.10: R-Z model of 1000 MWe fast breeder reactor core used for the study (control rods i.e. CSR and DSR are not considered in the model)

The exclusion of CSR and DSR from the model for optimization allows varying the number of subassemblies in different enrichment zones of the core in an easier way. Further, the presence of CSR and DSR are not having significant influence on the objectives and the final solutions of the optimization problem.

5.5.3. Mathematical Model Formulation of the Optimization Problem

The optimization problem presented in the study aims to find the optimal number of subassemblies in the three enrichment zones of a 1000 MWe fast breeder reactor core. The three enrichment zones are referred to as core-1, core-2 and core-3. The optimal number of subassemblies in the three enrichment zones are arrived while trying to satisfy the given objectives and constraints. The given optimization problem has six objectives and six constraints. The objectives for maximization are related to core excess reactivity (denoted by *RHO*) and breeding ratio (denoted by *BR*). The objectives for minimization are linear heat ratings of core-1 (denoted by *LHR*1), core-2 (denoted by *LHR*2), core-3 (denoted by *LHR*3) and percentage deviation of fuel inventory from a selected upper limit value (denoted by *FUI*). The upper and lower limits are defined for the constraints related to the parameters of *RHO*, *LHR*1, *LHR*2 and *LHR*3. The constraint related to *FUI* has an upper limit and that of *BR* has a lower limit. The limits of the constraints are taken in accordance with the uncertainties involved in their estimation. A solution to the problem can be termed as feasible, only if it satisfies all the six constraints. Accordingly, the mathematical formulation of the given optimization problem is given as:

Max (*RHO*, *BR*) and *Min* (*LHR*1, *LHR*2, *LHR*3, *FUI*)

= *f* (number of subassemblies of core-1, number of subassemblies of core-2, number of subassemblies of core-3) (5.17)

Such that,

$$8800 \le RHO \le 9200 \text{ pcm}, \; 485 \le LHR1 \le 505 \text{ W/cm},$$

$$510 \le LHR2 \le 530 \text{ W/cm}, 500 \le LHR3 \le 520 \text{ W/cm},$$

$$FUI < \text{given upper limit (in \% deviation), and } BR > 1.35 \qquad (5.18)$$

where, *Max* represents the maximization, *Min* represents the minimization and $f()$ represents "function of". The given objectives are function of the number of subassemblies of core-1 core-2 and core-3. The number of subassemblies explored for core-1, core-2 and core-3 are fixed to certain ranges, calculated based on the results from the initial trial runs of the neutronics simulation codes. By considering the ranges arrived, the optimization problem is assigned with three boundary conditions for the input values, as given below:

$$60 \le \text{number of subassemblies of core-1} \le 100 \qquad (5.19)$$

$$70 \le \text{number of subassemblies of core-2} \le 110 \qquad (5.20)$$

$$50 \le \text{number of subassemblies of core-3} \le 90 \qquad (5.21)$$

5.5.4. Model Formulation for the Penalty-function GA

The optimization model formulation of Penalty-function GA for the present study is similar to that of 500 MWe core, but has one more enrichment zone i.e. core-3. Therefore, there is an additional objective and constraint that is related to the linear heat rating of core-3 (denoted as *LHR*3). Among the six objectives of the problem, the maximization of *BR* is taken as the primary objective for the Penalty-function GA and the other five objectives are converted to penalty functions. The penalized objective function for the selected problem is formulated as follows:

$$Fitness = BR - P1 - P2 - P3 - P4 - P5 \qquad (5.22)$$

where,

$P1 \quad = |LHR1_{mid} - LHR1| \times A1,$ if $LHR1 < LHR1_{lb}$ or $LHR1 > LHR1_{ub}$

$\quad\quad = [LHR1 - (LHR1_{ub} + 1)] \times A1,$ otherwise

$P2 \quad = |LHR2_{mid} - LHR2| \times A2,$ if $LHR2 < LHR2_{lb}$ or $LHR2 > LHR2_{ub}$

$\quad\quad = [LHR2 - (LHR2_{ub} + 1)] \times A2,$ otherwise

$P3 \quad = |LHR3_{mid} - LHR3| \times A3,$ if $LHR3 < LHR3_{lb}$ or $LHR3 > LHR3_{ub}$

$\quad\quad = [LHR3 - (LHR3_{ub} + 1)] \times A3,$ otherwise

$P4 \quad = |RHO_{mid} - RHO| \times A4,$ if $RHO < RHO_{lb}$ or $RHO > RHO_{ub}$

$\quad\quad = [(RHO_{ub} + 1) - RHO] \times A4,$ otherwise

$P5 \quad = (FUI - FUI_{ub}) \times A5$

$P1$: penalty function related to $LHR1$

$P2$: penalty function related to $LHR2$

$P3$: penalty function related to $LHR3$

$P4$: penalty function related to RHO

$P5$: penalty function related to FUI

$A1$, $A2$, $A3$, $A4$ and $A5$: constant values selected to give proper weightage to the corresponding penalty functions

Fitness: penalized objective function used for the fitness evaluation in GA.

In the above equations, the subscripts have the following meanings related to the corresponding objective functions:

mid : denotes the middle value of the feasible range

lb : denotes the lower bound value of the feasible range

ub : denotes the upper bound value of the feasible range

The penalty functions (*P*1, *P*2, *P*3, *P*4 and *P*5) are formulated in such a way that, if the objectives fall with in the corresponding feasible range, then a positive value is added to the *BR* to get a higher "*Fitness*" value. On the other hand, if the objectives fall above or below the feasible range, then a negative value is added to the *BR* to get a lower "*Fitness*" value.

5.5.5. Model Formulation for the Multi-objective GA

The optimization model formulation of Multi-objective GA (for the NSGA-II implementation) is similar to that of 500 MWe core, but additionally has to handle the objective and the constraint of *LHR*3. The constraint violations are handled by an approach which is similar to the penalty handling mechanism in the Penalty-function GA. The constraint functions are first normalized and then the violation for each constraint is calculated. For the six constraints of the present study, corresponding constraint violations are calculated as:

$$C1 = \begin{cases} \dfrac{RHO - RHO_{mid}}{RHO_{min} - RHO_{mid}}, & \text{if } RHO < RHO_{lb} \\ \dfrac{RHO - RHO_{mid}}{RHO_{max} - RHO_{mid}}, & \text{if } RHO > RHO_{ub} \\ 0, & \text{otherwise} \end{cases} \qquad (5.23)$$

$$C2 = \begin{cases} \dfrac{LHR1 - LHR1_{mid}}{LHR1_{min} - LHR1_{mid}}, & \text{if } LHR1 < LHR1_{lb} \\ \dfrac{LHR1 - LHR1_{mid}}{LHR1_{max} - LHR1_{mid}}, & \text{if } LHR1 > LHR1_{ub} \\ 0, & \text{otherwise} \end{cases} \qquad (5.24)$$

$$C3 = \begin{cases} \dfrac{LHR2 - LHR2_{mid}}{LHR2_{min} - LHR2_{mid}}, & \text{if } LHR2 < LHR2_{lb} \\ \dfrac{LHR2 - LHR2_{mid}}{LHR2_{max} - LHR2_{mid}}, & \text{if } LHR2 > LHR2_{ub} \\ 0, & \text{otherwise} \end{cases} \qquad (5.25)$$

$$C4 = \frac{LHR3 - LHR3_{mid}}{LHR3_{min} - LHR3_{mid}}, \quad \text{if } LHR3 < LHR3_{lb}$$
$$= \frac{LHR3 - LHR3_{mid}}{LHR3_{max} - LHR3_{mid}}, \quad \text{if } LHR3 > LHR3_{ub}$$
$$= 0, \quad \text{otherwise} \qquad (5.26)$$

$$C5 = \frac{FUI - FUI_{u}}{FUI_{max} - FUI_{u}}, \quad \text{if } FUI > FUI_{u}$$
$$= 0, \quad \text{otherwise} \qquad (5.27)$$

$$C6 = \frac{BR_{l} - BR}{BR_{l} - BR_{min}}, \quad \text{if } BR < BR_{l}$$
$$= 0, \quad \text{otherwise} \qquad (5.28)$$

where, the terms $C1$, $C2$, $C3$, $C4$, $C5$ and $C6$ represents the constraint violation values related to *RHO*, $LHR1$, $LHR2$, $LHR3$, *FUI* and *BR* respectively. In the above equations, the subscripts have the following meanings related to the corresponding objective functions:

min	: minimum value possible
max	: minimum value possible
mid	: middle value of the feasible range
lb	: lower bound value of the feasible range
ub	: upper bound value of the feasible range
l	: lower limit value
u	: upper limit value

After calculating the six constraint violations, the overall constraint violation (C_{tot}) is calculated as:

$$C_{tot} = C1+C2+C3+C4+C5+C6 \qquad (5.29)$$

The next step is to modify the values of the objectives, according to the overall constraint violation. The overall constraint violation is multiplied with suitable constant values and the product is added to each of the objectives to get the modified values of the objective functions, as given below:

$$RHO_{mod} = RHO + B1 \times C_{tot} \tag{5.30}$$

$$LHR1_{mod} = LHR1 + B2 \times C_{tot} \tag{5.31}$$

$$LHR2_{mod} = LHR2 + B3 \times C_{tot} \tag{5.32}$$

$$LHR3_{mod} = LHR3 + B4 \times C_{tot} \tag{5.33}$$

$$FUI_{mod} = FUI + B5 \times C_{tot} \tag{5.34}$$

$$BR_{mod} = BR + B6 \times C_{tot} \tag{5.35}$$

where, the term C_{tot} represents the overall constraint violation. The subscript '*mod*' denotes the modified objective function values obtained. The terms $B1$, $B2$, $B3$, $B4$, $B5$ and $B6$ are used to denote constant values, assigned to make both terms on the right side of the above equations to have the same order of magnitude.

For a feasible solution, C_{tot} should be 0 and in that case, the modified values of the objective functions are same as that of actual objective function values. For an infeasible solution, a penalty is added to each of the objectives, in accordance with their constraint violations. Once the modified objective functions are calculated, those values are used by the multi-objective GA for Pareto-optimal fronts sorting.

5.5.6. Details of the Implementations

Implementation strategy followed in the present study is similar to that employed in the optimization study of 500 MWe core. The implementations of algorithms (both for the Penalty Function-GA and for the Multi-objective GA) are carried out by using the ‘C’ programming language and the interface module is implemented using the ‘R’ programming language. The same parameters of GA are selected for the implementation of both Penalty-function GA and Multi-objective GA.

There is an additional chromosome element for the present study, in order to represent the number of subassemblies in core-3. The presence of the additional chromosome element increases the computational complexity of the optimization problem and that is addressed in the implementation by increasing the population size. The population size for the present study is increased to 60 (for the study of 500 MWe core, the population size was 40).The study has been carried out on a computer system with Intel Core2 Duo CPU@3GHz and 2 GB RAM. The values assigned for the GA parameters are given in Table 5.6.

The two GA methodologies are implemented as per the details given above. The results of the study are presented next.

Table 5.6: Genetic parameters and methods or values used in the study. The parameters given in the table are kept the same for Penalty-function GA and Multi-objective GA.

Parameter	Methods/Values
Encoding	Floating point
Population size	60
Crossover Method	Arithmetical
Crossover Probability (CR)	0.6
Mutation Method	Non-uniform
Mutation Probability (MR)	0.025
Maximum no. of Generations	50

5.6. RESULTS OF THE STUDY OF 1000 MWe CORE CONFIGURATION

The procedures followed in generating the results are similar to that of the study of 500 MWe core. Several trial runs were conducted with randomly generated initial population of the GA (both for the Penalty Function-GA and for the Multi-objective GA). The fine-tuned versions of the algorithms are used to generate final results by conducting ten trial runs for each of the algorithms.

As mentioned earlier, the core configuration of the 1000 MWe fast breeder reactor designed by IGCAR is considered as the reference for the present study (see Section 5.1.1 for the description about the reference core). The number of fuel subassemblies designed for the reference core is compared with the results obtained from the optimization study, and presented in Table 5.7. The average of the number of subassemblies arrived from ten trial runs for each of the algorithms is furnished in the table. The values given in the table are arrived at, by rounding the average values (which are real numbers) to the nearest integers. It can be observed that both of the GA methodologies, i.e. the Penalty Function-GA and the Multi-objective GA, are able to generate feasible solutions which are agreeing with the number of subassemblies of the reference core.

Table 5.7: Comparison of the results obtained from the present study with the number of fuel subassemblies of the reference core (core of 1000 MWe Fast Breeder Reactor).

Fuel enrichment zones	Number of subassemblies		
	Reference core	Results of the present study	
		Penalty-function GA	Multi-objective GA
Core-1	79	79	78
Core-2	96	94	97
Core-3	72	72	72

5.6.1. Convergence of Objective Functions : Final Generation

The optimization problem presented in the study has six objectives and six constraints. The maximum and minimum values of the objectives, from the final generation of the Penalty Function-GA and the Multi-objective GA, are presented in Tables 5.8 and 5.9, respectively. Corresponding number of subassemblies arrived at, as the solutions to the optimization problem, are also given in the table.

The consolidated results, by considering the feasible solutions generated by the Penalty Function-GA and the Multi-objective GA, in the final generations of ten trial runs are furnished in Table 5.10. The maximum (Max), minimum (Min), average (Ave) and standard deviation (SD) values for the six objectives are given in the table. The maximum values are calculated by taking the average of maximum values produced (separately for each of the six objective functions) at the final generation for the ten trial runs. Similarly, the minimum values are calculated by taking the average of minimum values generated. The average values shown in the table are calculated by finding the average values of respective objectives (by considering the 50 members of the final population) for the ten trial runs and then, the corresponding standard deviations are calculated.

The convergence of the six objectives of the optimization problem, taken up in the study can be clearly pictured using box plots as shown in Figure 5.11. The results generated at the final generation (i.e. 50^{th} generation) of the Penalty-function GA (Figure 5.11(a)) and the Multi-objective GA (Figure 5.11(b)), by the ten trial runs are given in the figures. The plots shows that the overall convergences of the objectives are similar to the convergences of the objectives presented for the study of 500 MWe core (see Figure 5.5).

Table 5.8: Maximum and minimum values obtained in the final generation for the six objective functions and the corresponding solutions (Penalty function-GA).

Tr. No.	Max /Min	Values obtained for objective functions						Solutions arrived		
		RHO	*LHR*1	*LHR*2	*LHR*3	*FUI*	*BR*	Core-1	Core-2	Core-3
1	Max	8968.17	501.52	524.48	517.62	-16.98	1.3562	79	97	72
	Min	8919.88	496.39	519.90	513.22	-16.59	1.3550	77	95	71
2	Max	8926.59	504.31	524.37	518.07	-17.36	1.3580	79	94	72
	Min	8846.15	498.67	521.43	517.09	-16.94	1.3561	79	94	71
3	Max	8926.59	500.50	522.98	521.05	-17.25	1.3565	79	94	72
	Min	8916.28	498.67	521.43	518.07	-16.94	1.3561	79	93	72
4	Max	8968.17	501.98	524.48	517.62	-17.01	1.3569	78	97	72
	Min	8888.41	496.39	520.69	513.22	-16.59	1.3550	77	95	71
5	Max	8926.59	504.82	523.53	521.05	-17.39	1.3588	80	94	72
	Min	8814.78	498.67	520.64	517.53	-16.94	1.3561	79	93	71
6	Max	8926.59	504.21	526.10	527.10	-17.89	1.3572	79	94	72
	Min	8895.73	498.67	521.43	518.07	-16.94	1.3561	79	91	72
7	Max	8926.59	499.15	521.43	518.50	-16.97	1.3568	80	94	72
	Min	8895.43	497.31	519.11	515.54	-16.65	1.3561	79	93	72
8	Max	8955.22	499.15	520.64	519.84	-16.97	1.3571	81	94	73
	Min	8874.81	492.22	515.46	515.54	-16.25	1.3552	80	92	72
9	Max	8926.59	506.19	525.92	521.05	-17.68	1.3585	79	94	72
	Min	8835.29	498.67	521.43	518.07	-16.94	1.3561	79	93	71
10	Max	8926.59	500.99	522.18	521.49	-17.28	1.3572	80	94	72
	Min	8885.01	497.31	519.11	515.54	-16.65	1.3561	79	92	72

Table 5.9: Maximum and minimum values obtained in the final generation for the six objective functions and the corresponding solutions (Multi-objective GA).

Tr. No.	Max /Min	Values obtained for objective functions						Solutions arrived		
		RHO	*LHR*1	*LHR*2	*LHR*3	*FUI*	*BR*	Core -1	Core -2	Core -3
1	Max	9160.96	504.84	528.41	519.84	-17.59	1.3587	83	101	73
	Min	8805.57	486.19	510.07	500.06	-15.07	1.3501	75	92	70
2	Max	9157.59	504.84	526.79	519.72	-17.36	1.3590	83	104	73
	Min	8802.46	485.18	510.74	500.20	-15.00	1.3505	72	90	70
3	Max	9179.71	504.36	529.38	519.68	-17.55	1.3588	86	103	73
	Min	8802.46	485.58	510.18	500.06	-15.04	1.3501	71	88	71
4	Max	9179.71	504.84	529.33	519.84	-17.59	1.3583	81	102	73
	Min	8816.62	485.58	510.97	500.01	-15.04	1.3501	71	92	70
5	Max	9097.23	504.82	526.03	519.87	-17.39	1.3588	84	96	74
	Min	8805.57	485.79	510.18	504.79	-15.30	1.3518	77	92	71
6	Max	9151.38	504.84	528.49	518.09	-17.24	1.3584	79	101	70
	Min	8811.47	485.18	512.47	500.48	-15.18	1.3504	74	96	73
7	Max	9128.13	504.78	529.32	519.68	-17.59	1.3584	82	100	74
	Min	8807.96	485.36	510.07	500.06	-15.04	1.3510	73	95	70
8	Max	9100.83	503.83	525.20	519.84	-17.33	1.3587	83	99	74
	Min	8805.57	485.41	510.21	501.10	-15.07	1.3519	76	91	71
9	Max	9077.53	504.78	529.33	519.68	-17.59	1.3580	82	102	73
	Min	8836.71	485.58	510.07	500.06	-15.00	1.3519	73	94	70
10	Max	9141.79	504.82	528.41	519.68	-17.59	1.3588	82	99	73
	Min	8807.96	486.92	511.54	500.06	-15.07	1.3508	75	93	71

Table 5.10: Values of feasible solutions obtained in the final generation (for ten trial runs) of Penalty-function GA and Multi-objective GA.

		Max	Min	Ave	SD
Penalty-function GA	*RHO*	8937.77	8877.18	8900.87	20.95
	*LHR*1	502.28	497.30	500.25	1.67
	*LHR*2	523.61	520.06	522.10	1.07
	*LHR*3	520.34	516.19	518.56	1.31
	FUI	-17.28	-16.74	-17.09	-99.83
	BR	1.3573	1.3558	1.3567	0.0005
Multi-objective GA	*RHO*	9137.49	8810.24	8946.76	56.83
	*LHR*1	504.67	485.68	495.63	5.21
	*LHR*2	528.07	510.65	519.03	5.03
	*LHR*3	519.59	500.69	510.19	5.39
	FUI	-17.48	-15.08	-16.26	-99.31
	BR	1.3586	1.3509	1.3553	0.0014

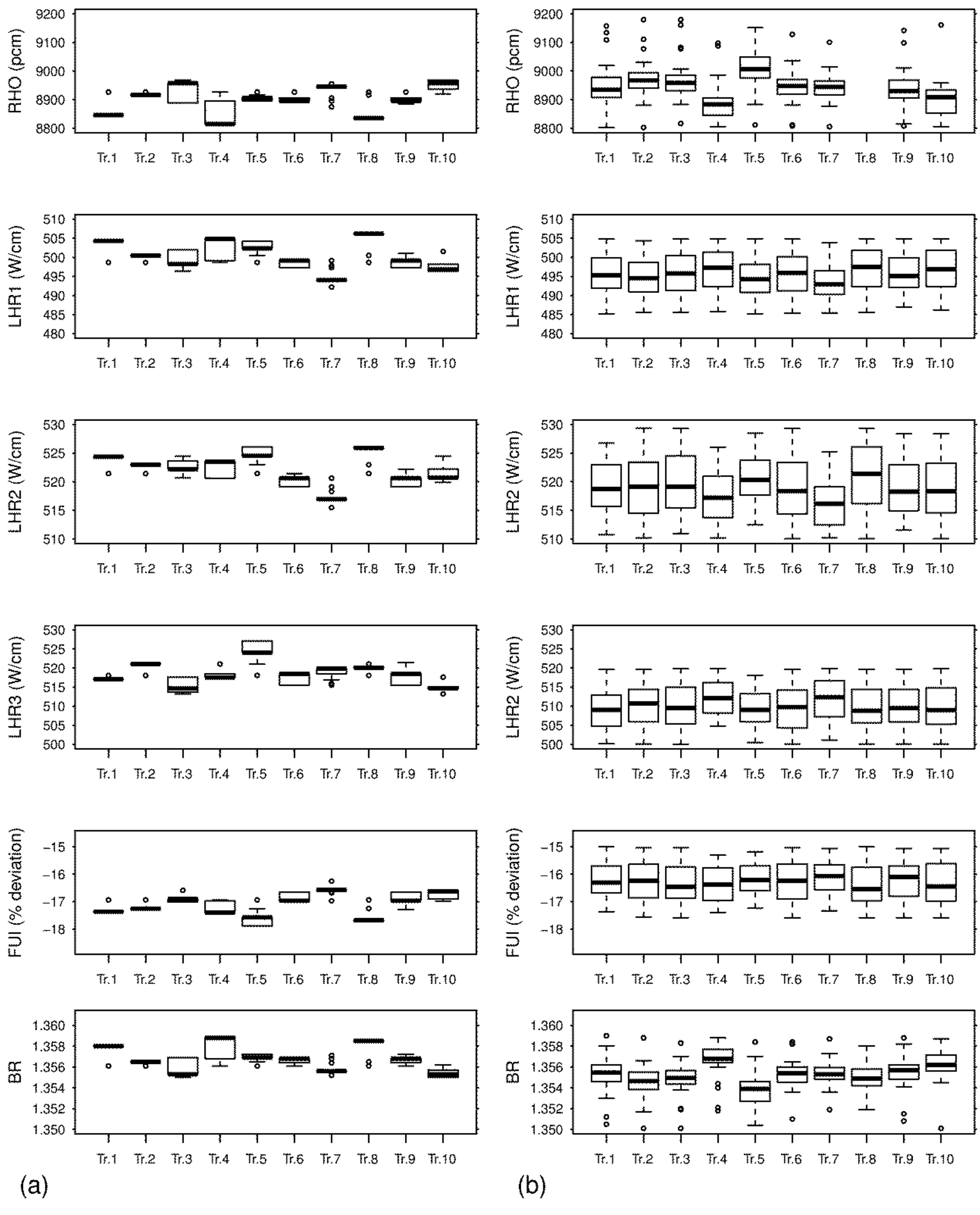


Figure 5.11: Box plots to represent the five objective functions distribution among the generated feasible solutions in final generation of ten trial runs: (a) for Penalty-function GA. (b) for Multi-objective GA

The major observations from the results presented in Tables 5.8 to 5.10 and Figure 5.11 are:

- Both of the algorithms, i.e. the Penalty Function-GA and the Multi-objective GA, are able to produce feasible solutions consistently at the final generation.
- The feasible solutions generated by the Penalty Function-GA remains in a narrower range as compared with the Multi-objective GA.

It is important to note that the observations mentioned above are similar to the observations from the study of 500 MWe core.

5.6.2. Comparison of GA Performance: Generation wise Production of Feasible Solutions

The performances of the algorithms are compared based on the number of feasible solutions produced in successive generations. In the present problem, one generated solution is termed as feasible, if it satisfies all the six constraints. That is, a feasible solution should satisfy the conditions:

(i) *RHO* should be in the range between 8800 and 9200 pcm

(ii) *LHR*1 should be in the range between 485 and 505 W/cm

(iii) *LHR*2 should be in the range between 510 and 530 W/cm

(iv) *LHR*3 should be in the range between 500 and 520 W/cm

(v) *FUI* should be less than the given upper limit

(vi) *BR* should be greater than 1.35

Figure 5.12 shows the average number of feasible solutions produced in successive generations by ten trial runs of the Penalty-function GA (Figure 5.12(a)) and ten trial runs of the Multi-objective GA (Figure 5.12(b)). The conditions of feasibility are kept the same for both the algorithms. The average CPU times and generations taken for producing 80% of population with feasible solutions (i.e. 48 feasible solutions out of the 60 members of the population), are taken in the comparison. The Penalty-function GA took 23 generations and 3937.37 seconds to produce 48 feasible solutions.

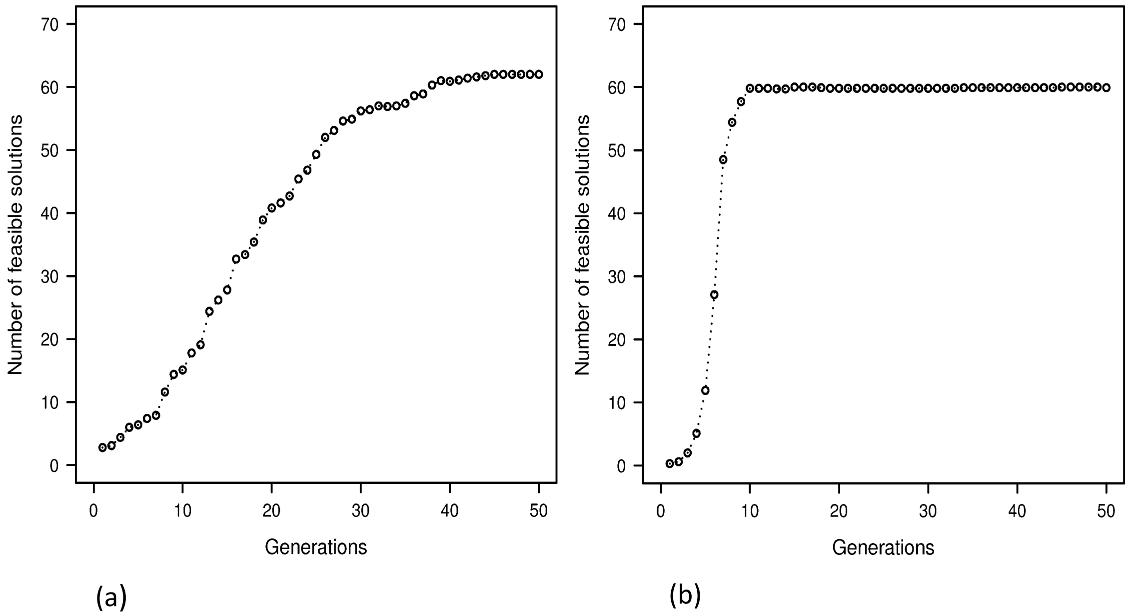


Figure 5.12: Average Number of feasible solutions (generation wise) of the ten trial runs: (a) for Penalty-function GA. (b) for Multi-objective GA.

The Multi-objective GA took 8 generations and 1328.00 seconds to produce the same number of 48 feasible solutions; implying that the Multi-objective GA is 66% faster than Penalty-function GA, with respect to CPU time for generating 80% of the population with feasible solutions. The behavior observed, can be utilized in reducing the computational time by lowering the number of required generations of the algorithm. Another observation is that, when the computational time requirement for a fixed number of generations are considered, there is no significant difference between Penalty-function and Multi-objective GA. The average CPU time requirement (for ten trial runs, each with 50 generations) for Penalty-function GA is 8559.29 seconds whereas for Multi-objective GA it is 8300.28 seconds. It is seen that, when average computational time of fixed generations are considered, Multi-objective GA has a marginal advantage of 3% faster than Penalty-function GA.

The overall observation from the comparison is that the Multi-objective GA has a significant advantage in convergence speed and has a marginal advantage in computational time, when compared with the Penalty-function GA.

5.6.3. Generation wise Evolution of Objective Functions

The convergence of the six objectives during the whole evolution of the Penalty-function GA and the Multi-objective GA is presented in Figure 5.13. The solid black points denote the average value of the objective and the dotted lines above and below denote the standard deviation. The convergences of the six objectives for the Penalty-function GA are shown in Figures 5.13(a) to 5.13(f). Similarly, the convergences of the six objectives of Multi-objective GA are shown Figures 5.13(g) to 5.13(l). The observations from comparing the results are given below:

- Both of the algorithms, i.e. Penalty-function GA and Multi-objective GA, are capable of arriving at around the same converged area of feasible solutions with respect to the given objectives.
- The objective functions converge to a narrower region towards the end of generations for the Penalty-function GA as compared to the Multi-objective GA.
- The speed of convergence is faster for the Multi-objective GA as compared to the Penalty-function GA.

5.6.4. Solution Diversity: Initial Population Vs. Final Population

A solution candidate of the given optimization problem represents the number of subassemblies present in core-1, core-2 and core-3. The distribution of solution candidates (of a single trial run) in the population of initial generation and in the population of final generation are compared using the 3-D plots shown in Figure 5.14

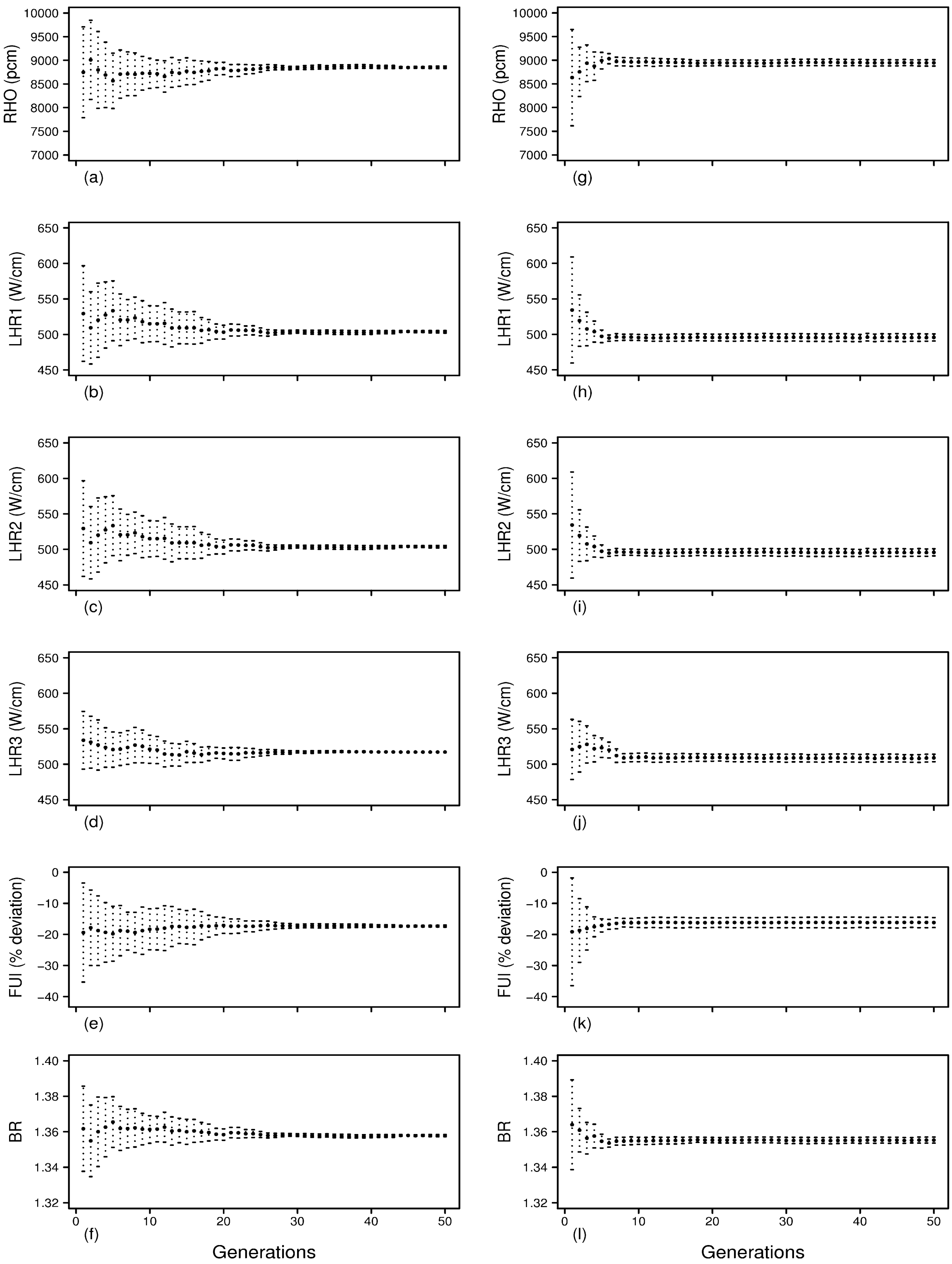


Figure 5.13: Convergences of the objectives during the generation wise evolution of the algorithms: (a) to (e) for Penalty-function GA. (f) to (j) for Multi-objective GA.

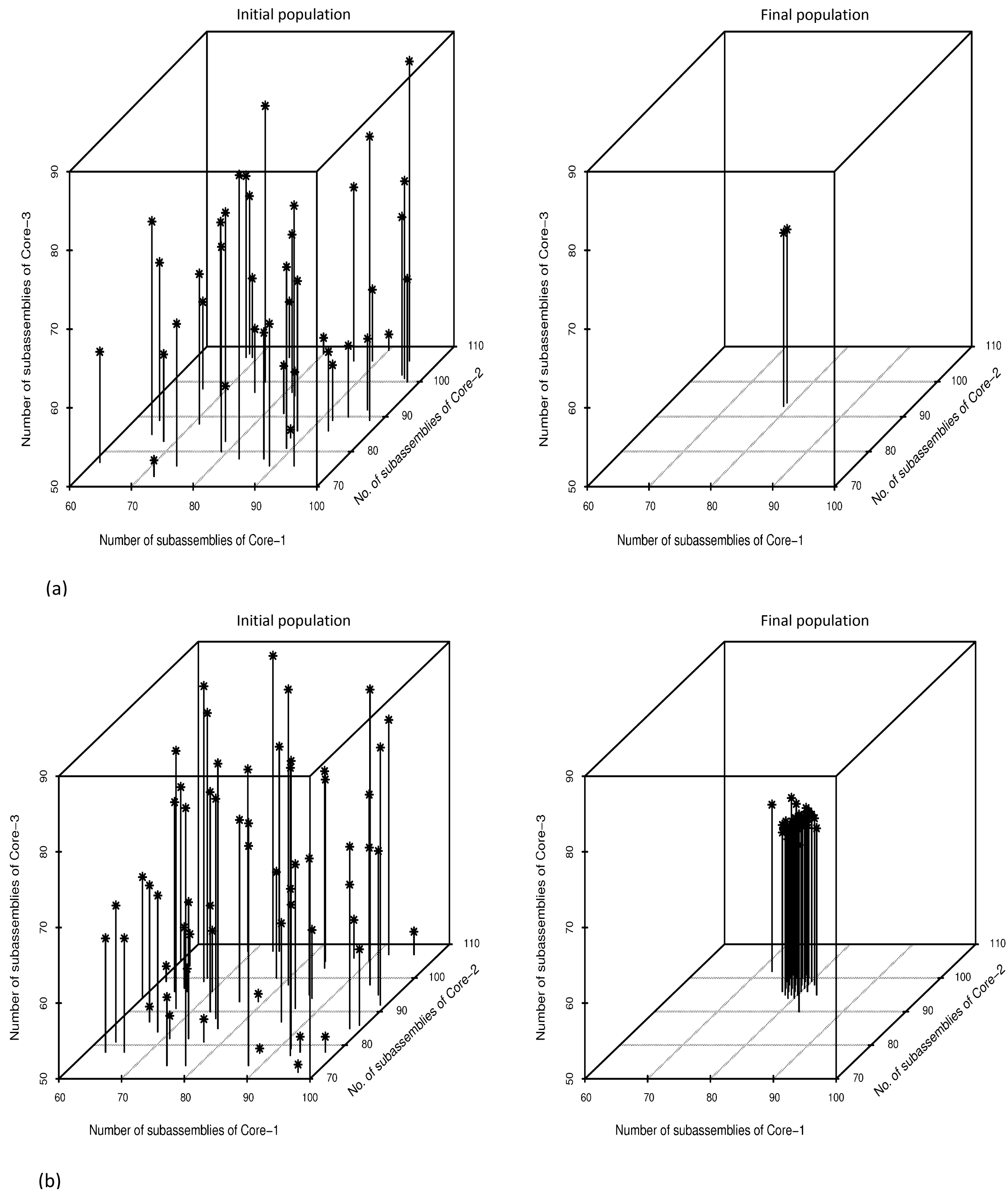


Figure 5.14: Comparison of solution diversity of the algorithms (initial population vs. final population) (a) for Penalty-function GA. (b) for Multi-objective GA.

The star shaped heads of the vertical lines represent the solution candidates in the figure. The solution diversity of the initial and the final populations for the Penalty-function GA is shown in Figure 5.14(a) and that of the Multi-objective GA is shown in Figure 5.14(b). As shown in the figure, the solution candidates converge to a narrow area for the Penalty-function GA, but to a wider area for the Multi-objective GA. Therefore, as in the case of diversity among the objectives, the diversity among the solution candidates is also better for the Multi-objective GA.

5.7. SUMMARY

Two separate optimization studies of core configuration, based on the cores of two different fast breeder reactors, are included in this chapter. The first study is for a 500 MWe core with two fuel enrichment zones and the second study is for a 1000 MWe core with three fuel enrichment zones. The aim of the studies is to find the optimal number of fuel subassemblies in the different enrichment zones of the cores. Two GA methodologies i.e., the Penalty-function GA and the Multi-objective GA are applied and compared.

The result obtained from the studies show that, both the Penalty-function GA and the Multi-objective GA are capable of arriving at, around the same converged area of feasible solutions, with respect to all the objectives. When diversity of the feasible solutions are compared, the Multi-objective GA performs better than the Penalty-function GA. The Multi-objective GA has better convergence speed also. When average computational times of fixed generations are considered, it is observed that both the Penalty-function GA and the Multi-objective GA perform almost equally well.

The modular approach followed in the burnup optimization study (presented in chapter 4) has been enhanced in the present studies, by employing the pattern searching capabilities of the ‘R’ programming language. The ‘R’ programming language is used for developing the interface module, which facilitates a smooth communication between the GA module (which is developed in ‘C’ programming language) and the neutronics simulation codes which are based on FORTRAN programming language. The changes in the GA modules can easily be done without affecting the interface module. This modular approach has given the advantage of extending the implementation easily to other studies of similar nature. The studies helped in establishing a path for the application of GA in many other optimization problems of the core design, pertaining to the fast breeder reactor programme.

CHAPTER 6

SUMMARY AND SUGGESTIONS FOR FUTURE STUDIES

This chapter presents summary and important conclusions drawn from the results of the research work, carried out in the domain of the application of Genetic Algorithm based optimization methodologies to some diverse reactor environments. Suggestions for extending the applications of such intelligent optimization methodologies to other reactor systems are also given.

6.1. SUMMARY

The motivation behind the work carried out in the present study was the application and comparison of Genetic Algorithm based optimization methodologies in different reactor environments, with the aim of arriving at the suitability of a given methodology for a particular type of application. The possibility of developing a modular approach in the implementations of the methodologies was explored. The methodologies applied and studied are standard real-parameter GA, Penalty-function GA and Multi-objective GA. The subsystems of reactors selected are:

(i) steam condenser of Prototype Fast Breeder Reactor (PFBR)

(ii) burnup zones of the core of a 220 MWe Pressurized Heavy Water Reactor (PHWR)

(iii) fuel enrichment zones of the cores of a 500 MWe Fast Breeder Reactor (FBR)

(iv) fuel enrichment zones of the cores of a 1000 MWe FBR

The study on steam condenser is an engineering design optimization and the other three studies pertain to nuclear fuel management. The chapter wise content of the thesis is summarized below:

- An introduction to the research work has been presented in Chapter 1. The description and findings from the literature survey carried out in the field of Computational Intelligence methods applicable in different nuclear subsystems, including the applications of the GA in nuclear fuel management, were given. The GA methodologies employed for nuclear fuel management and their features were summarized in the chapter. The objectives and the scope of the work carried out in the present study were described in the chapter.
- Chapter 2 introduced the Genetic Algorithm operators and parameters used in the studies carried out as part of the thesis. Several investigations were conducted in order to understand the influence of various representation schemes and operators on the performance of the algorithms, using a typical mathematical optimization function named Ackley's function. The two methodologies of GA which were employed in the optimization studies of nuclear fuel management, i.e. Penalty-function GA and Multi-objective GA, were discussed in detail. In essence, the tools and methods employed in different studies presented in the thesis are explained in the chapter.
- An optimization study for the engineering design, conducted on the steam condenser (circulating water system) of Prototype Fast Breeder Reactor (PFBR), was discussed in Chapter 3. The purpose of the study was to apply standard real-parameter Genetic Algorithm for the single objective

optimization problem pertaining to the design of the circulating water system. In the study, the GA based performance-cost analysis was carried out based on the maximum capitalized profit generated by the system. The suitability of the standard real-parameter GA in the engineering design application of a reactor subsystem was demonstrated by the study.

- Chapter 4 described the work carried out for the fuel bundle burnup optimization of a Pressurized Heavy Water Reactor (PHWR). The aim of the study was to calculate the optimum discharge burnups of the fuel bundles in the inner and outer burnup zones which give maximum reactor power, by satisfying the multiple constraints. The purpose of the study was to apply and compare the performance of the Penalty-function GA and the Multi-objective GA in the burnup optimization problem having multiple objectives and constraints. The discharge burnups arrived by the GA based optimization procedure can be utilized in fixing the most suitable reference discharge burnups for the two burnup zones of the reactor core. A modular approach has been employed in the implementation of the optimization methodologies. The results obtained from the study showed the suitability of Multi-objective GA over Penalty-function GA, in solving problems like the one selected for this study.
- The problem of finding out optimal number of subassemblies in the core of Fast Breeder Reactor (FBR) was addressed in Chapter 5. This was with the aim of achieving the best performance. Two different Fast Breeder Reactor cores, (i) 500 MWe core having two different fuel enrichment zones (ii) 1000 MWe core having three different fuel enrichment zones, were considered in the studies. Two optimization methodologies, i.e. the

Penalty-function GA and the Multi-objective GA have been applied and compared. The Multi-objective GA was found to be better than the Penalty-function GA, in terms of diversity of generated solutions and in the speed of convergence. The modular approach followed in the burnup optimization study (presented in chapter 4) has been enhanced by employing the pattern searching capabilities of the 'R' programming language.

6.2. SUMMARY OF THE RESULTS

The work carried out in the present study addresses three gap areas found in the current state of the art of optimization in nuclear fuel management using the GA. The gap areas are: (i) application of GA in nuclear fuel management of Pressurized Heavy Water Reactors and Fast Breeder Reactors are not explored in full potential (ii) a modular approach that divides total optimization procedure in to GA module, interface module and neutronics simulation module, is not available (iii) the organized way of application and comparison of GA based optimization methodologies, with the aim of verifying their suitability, is not addressed. The studies have given some contributions to bridge the gaps mentioned above. The major findings from the studies are summarized as follows:

- the selected GA methodologies namely, Penalty-function GA and Multi-objective GA, are suitable for solving nuclear fuel management problems of Pressurized Heavy Water Reactors and Fast Breeder Reactors
- The Multi-objective GA is found to be better than the Penalty-function GA in two important aspects: one, in terms of the diversity of generated solutions and two, in the speed of convergence of the algorithm.
- The Penalty-function GA is easier to model and implement than the Multi-objective GA. Therefore, when the diversity in the solutions is of less

importance and the savings in the time of the software development is of more importance, the Penalty-function GA becomes a good option.

- The suitability of standard real-parameter Genetic Algorithm is verified for the engineering design optimization of reactor subsystem with single objective and limited number of constraints.
- A modular approach has been employed, in applying and comparing different methodologies based on the GA, in the optimization problems of nuclear fuel management. The optimization procedure employed in the study was divided in to the GA module, the interface module and the neutronics simulation module. This approach has provided the possibility of developing more efficient communication mechanisms and the flexibility in modifying the modules for extending the applications to other reactor systems.
- The tools and methods employed for the comparison of the GA methodologies were uniformly applied to different types of fuel management problems, in an organized way. The parameters taken up for the comparisons of methodologies, pertaining to different studies carried out as the part of the thesis, are:
  (i) maximum and minimum values of the objective function
  (ii) diversity among the objective functions and among the feasible solutions
  (iii) generation wise production of the feasible solutions.

6.3. SUGGESTIONS FOR FUTURE STUDIES

The modular approach followed in the present studies, allow extending the application of the GA methodologies to many other optimization studies of nuclear fuel management. In the direction of extending the applications, the following areas of study can be considered:

(i) fuel bundle optimization of Pressurized Heavy Water Reactor (PHWR) core with different geometries of inner and outer burnup zones

(ii) finding out the optimal control rod positions in the Fast Breeder Reactor (FBR) core

(iii) fuel enrichment optimization in different zones of the FBR core

There are many other “nature inspired intelligent algorithms” such as Tabu Search, Ant Colony Optimization, Particle Swarm Optimization, Artificial Bee Colony Optimization, Harmony Search Algorithm, Continuous Firefly Algorithm etc. and the field is continuously evolving. Some of the methods mentioned above, have the advantage of reduced search length, resulting in less computational time, but perhaps at the price of a reduced capability of coming close to the true optimum. It is possible to try these methods and evaluate their suitability in the optimization studies of different reactor subsystems. Another possibility in this direction is the development and application of efficient “parallel” genetic algorithms (genetic algorithms developed by employing parallel programming concepts) which are capable of running in parallel computers.

The work presented in the current thesis explores the application of Genetic Algorithm, one of the major optimization methods coming under the umbrella of Computational Intelligence, to a set of diverse reactor environments. The role of Computational Intelligence in the domain of nuclear reactors is getting wider acceptance by considering the present objectives of the plant operation and the future needs. The development and use of such methods is important for the next generation nuclear reactors, which are expected to operate semi-autonomously for long periods of time.

REFERENCES

[1] Computational Intelligence Principles: Techniques and Applications
A. Konar
Springer International Edition Chapter 1(2005) 1-17

[2] Computational Intelligence: An Introduction, Second Edition
A.P. Engelbrecht
Jon Wiley & Sons Ltd Chapter 1(2007) 1-13

[3] Computational Intelligence A Logical Approach
D. Poole, A. Mackworth, R. Goebel
Oxford University Press (1998) Chapter 1, 1-20

[4] Computational Intelligence in Product Design Engineering: Review and Trends
A. Kusiak, F. A. Salustri
IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews 37(2007) 767-778

[5] A review and evalution of multi-objective algorithms for the flow shop scheduling problem
G. Minella, R. Ruiz, M. Ciavotta
INFORMS Journal on Computing 20(2008) 451-471

[6] Design and Control of Intelligent Robotic Systems
D. Liu, L. Wang, K.C. Tan
Springer (2009) Chapter 1, 1-20

[7] Autonomous Visual Mapping and Exploration With a Micro Aerial Vehicle
L.Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, F. Fraundorfer, M. Pollefeys
Journal on Field Robotics 31(2014) 654-675

[8] Business Intelligence and Analytics : From Big Data to Big Impact
H. Chen, Roger H. L. Chiang, Veda C. Storey
MIS Quarterly, 36(2012) 1165-1188

[9] Jumping NLP Curves: A Review of Natural Language Processing Research
E. Cambria, B. White
IEEE Computational Intelligence Magazine (2014) 48-57

[10] A real time expert system model for emergency control of modern power systems
S.S. Kulkarni, S.A. Khaparde, D.G. Bharadwaj, G. Sivakumar
Systems Science 12(1994)

[11] Extracting decision trees from trained neural networks
R. Krishnan, G. Sivakumar, P. Bhattacharya
Pattern Recognition 32(1999) 1999-2009

[12] A study of the bipartite ranking problem in machine learning
S. Agarwal
PhD dissertation, University of Illinois at Urbana-Champaign (2005)

[13] Ranking genes by relevance to a disease
S. Agarwal, S. Sengupta
Proceedings of the 8th International Conference on Computational Systems Bioinformatics (CSB) (2009)

[14] http://www.world-nuclear.org
World Nuclear Association (accessed 03-July-2015).

[15] Nuclear Reactor Engineering: Reactor design basics
S. Glasstone, A. Sesonske
Springer 4th edition (1994)

[16] Fast neutron reactors: principal features and experience
A. Rineiski
Workshop on fusion for neutrons and sub-critical nuclear fission
FUNFI, Italy, (September 12-15, 2011)

[17] Nuclear Power: Thorium for energy independence
A. Kakodkar
Electrical India (Oct. 2006) 85-87

[18] Thorium utilization for sustainable supply of nuclear energy
S. Banerjee, R.K. Sinha, S. Kailas
Journal of Physics: Conference Series 312(2011)

[19] Development of Computational Intelligence Systems for parameter estimation and event identification in Fast Breeder Reactor
R.P. Subhra
PhD dissertation, Homi Bhabha National Institute, Mumbai, India
Chapter 1(2012) 5-10

[20] Neural networks and statistical techniques: A review of applications
M. Paliwal, U.A. Kumar
Expert Systems with Applications 36(2009) 2-17

[21] Transient identification in nuclear power plants: A review
K.M. Bakhshayesh, and M.B. Ghofrani
Progress in Nuclear Energy 67(2013) 23-32

[22] Nuclear Power Plant Status Diagnostics Using an Artificial Neural Network
E.B. Bartlett, R.E. Uhrig
Nuclear Technology 97(1992) 272-281

[23] A Dynamic Neural Network Aggregation Model for Transient Diagnosis in Nuclear Power Plants
K. Mo, S.J. Lee, P.H. Seong
Progress in Nuclear Energy 49(2007) 262-272

[24] Power Prediction in Nuclear Power Plants Using a Back-Propagation Learning Neural Network
M.S. Roh, W.S. Cheon, S.H. Chang
Nuclear Technology 94(1991) 270-278

[25] Adaptive Control of a PWR Core Power Using Neural Networks
H. Arab-Alibeik, S. Setayeshi
Annals of Nuclear Energy 32(2005) 588-605

[26] Computational Intelligence in Nuclear Engineering
R.E. Uhrig, J.W. Hines
Nuclear Engineering and Technology 37(2005) 127-138

[27] Traditional signal pattern recognition versus artificial neural networks for nuclear plant diagnostics
S. Keyvan
Progress in Nuclear Energy 39(2001) 1-29

[28] Elman's recurrent neural network applications to condition monitoring in nuclear power plant and rotating machinery
S. Seker, E. Ayaz, E. Turkcan
Engineering Applications of Artificial Intelligence 16(2003) 647-656

[29] Fuzzy logic systems for engineering: a tutorial
J. Mendel
Proceedings of IEEE 83(1995) 345-377

[30] Fuzzy Power Control Algorithm for a Pressurized Water Reactor
Y.J. Hah, B.Y. Lee
Nuclear Technology 106(1994) 242-253

[31] Design and optimization of fuzzy-PID controller for the nuclear reactor power control
C. Liu, J.F. Peng, F.Y. Zhao, C. Li
Nuclear Engineering and Design 239(2009) 2311-2316

[32] A Stable Adaptive Fuzzy Control Scheme for Tracking an Optimal Power Profile in a Research Nuclear Reactor
E.R. Ramirez, Jorge S. Benitez-Read, Armando Segovia-De-Los Rio
Annals of Nuclear Energy 58(2013) 238-245

[33] A Fuzzy-Gain-Scheduled Neural Controller for Nuclear Steam Generators
H. Habibiyan, S. Setayeshi, H. Arab-Alibeik
Annals of Nuclear Energy 31(2004) 1765-1781

[34] Fuzzy Logic Control of Steam Generator Water Level in Pressurized Water Reactors
C.C. Kuan, C. Lin, C.C. Hsu
Nuclear Technology 100(1992) 125-134

[35] Design of a Fuzzy Logic Controller for Water Level Control in an Advanced Boiling Water Reactor Based on Input-Output Data
C. Lin, D.H. Yang
Nuclear Technology 122(1998) 318-329

[36] Genetic Algorithms and Engineering Design
M. Gen, R. Cheng
Wiley-Interscience Publication, John Wiley & Sons (1997)

[37] Genetic Algorithms for Estimating Effective Parameters in a Lumped Reactor Model for Reactivity Predictions
M. Marseguerra, E. Zio
Nuclear Science and Engineering 139(2001) 96-104

[38] Optimization of the Power Ascension Path for a Boiling Water Reactor Using Genetic Algorithms
C.S. Lee, C. Lin
Annals of Nuclear Energy 33(2006) 53-61

[39] Automatic Search of Power Ascension Path for a Boiling Water Reactor Using Genetic Algorithm and Neural Network
C.S. Lee, C. Lin
Nuclear Technology 159(2007) 256-266

[40] Design of a model predictive load-following controller by discrete optimization of control rod speed for PWRs
J.H. Kim, S.H. Park, M. Gyun Na
Annals of Nuclear Energy 71(2014) 343-351

[41] A genetic algorithm solution for a nuclear power plant risk-cost maintenance model
T. Jiejuan, M. Dingyuan, X. Dazhi
Nuclear Engineering and Design 229(2004) 81-89

[42] Optimisation of ISI interval using genetic algorithms for risk informed in-service inspection
V. Gopika, H.S. Kushwaha, A.K. Verma, A. Srividya
Reliability Engineering and System Safety 86(2004) 307-316

[43] Parallel Island Genetic Algorithm Applied to a Nuclear Power Plant Auxiliary Feed water System Surveillance Tests Policy Optimization
C.M.N.A.Pereira, C.M.F. Lapa
Annals of Nuclear Energy 30(2003) 1665-1675

[44] Safety Management in NPPs Using an Evolutionary Algorithm Technique
A. Mishra, A. Patwardhan, A.K. Verma
Nuclear Engineering and Design 237(2007) 1445-1451

[45] Regulation of nuclear power plants a multi objective approach
A. Mishra, D. Pandey, A. Chauhan
Annals of Nuclear Energy 36(2009) 1560-151.112

[46] Advanced Progressive Real Coded Genetic Algorithm for Nuclear System Availability Optimization Through Preventive Maintenance Scheduling
M. Aghaie, A. Norouzi, A. Zolfaghari, A. Minuchehr, M.Z. Fard, R. Tumari
Annals of Nuclear Energy 60(2013) 64–72

[47] Design and Development of Genetic Algorithm for Test Interval Optimization of Safety Critical System for a Nuclear Power Plant
M. Mehra, M.L. Jayalal, A. John Arul, S. Rajeswari, K.K. Kuriakose, S.A.V.SatyaMurty
International Conference on Intelligent Systems Design and Applications (ISDA-2012) Kochi, Kerala, India November 27-29(2012)

[48] Study on Different Crossover Mechanisms of Genetic Algorithm for Test Interval Optimization for Nuclear Power Plants
M. Mehra, M.L. Jayalal, A. John Arul, S. Rajeswari, K.K. Kuriakose, S.A.V. Satya Murty
Intelligent Systems and Applications 1(2014) 20-28

[49] Genetic Algorithm applied to turbine extraction optimization of a pressurized-water reactor
W.F. Sacco, C.M.N.A. Pereira, P.P.M. Soares, R. Schirru
Applied Energy 73(2002) 217-222.

[50] Use of a genetic algorithm in the search for a near-optimal shielding design
B.S. Kim, J.H. Moon
Annals of Nuclear Energy 37(2010) 120-129

[51] Application of genetic algorithms to in-core nuclear management optimization P.W. Poon, G.T. Parks
Proceedings ofthe Joint International Conference on "Mathematical Methods and Super-computing in Nuclear Applications", (1993) Karls-ruhe, Germany, 777-786

[52] Advances in Nuclear Science and Technology
J. Lewins, M. Becker
Springer Volume 26(1999) Chapter 5 137-162

[53] Optimization of Thorium loading in fresh core of Indian PHWR by evolutionary algorithms
S. Mishra, R.S. Modak, S. Ganesan
Annals of Nuclear Energy 36(2009) 948-955

[54] Optimization of depleted uranium bundle loading in fresh core of Indian PHWR by evolutionary algorithm
S. Mishra, R.S. Modak, S. Ganesan
Annals of Nuclear Energy 37(2010) 208-217

[55] Optimization of depleted uranium loading in fresh core of large sized Indian PHWR by evolutionary algorithm
S. Mishra, R.S. Modak, S. Ganesan
Annals of Nuclear Energy 38(2011) 905-909

[56] Canada deuterium uranium reactor design optimization using three dimensional generalized perturbation theory
D. Rozon, M. Beaudet
Nuclear Science and Engineering 11(1982) 1-20

[57] Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence
Holland, H. John
Oxford, England: U Michigan Press (1975)

[58] Optimization by Simulated Annealing
S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi
Science 220(1983) 671–680

[59] BWR fuel reload design using a Tabu search technique
A. Castillo, G. Alonso, L.B. Morales, C.M. del Campo, J.L. Francois, E. del Valle
Annals of Nuclear Energy 31(2004) 151-161

[60] A New System to Fuel Loading and Control Rod Pattern Optimization in Boiling Water Reactors
J.J. Ortiz, A. Castillo, J.L. Montes, R. Perusquía
Nuclear Science and Engineering 157(2007) 236-244

[61] Automatic pressurized water reactor loading pattern designusing ant colony algorithms
C. Lin, B.F. Lin
Annals of Nuclear Energy 43(2012) 91-98.

[62] The Ant-Q algorithm applied to the nuclear reload problem
L. Machado, R. Schirru
Annals of Nuclear Energy 29(2002) 1455-1470

[63] Particle swarm optimization applied to the nuclear reload problem of a Pressurized water reactor
A.A.M. Meneses, M.D. Machado, R. Schirru
Progress in Nuclear Energy 51(2009) 319-326

[64] Multiprocessor modeling of parallel Particle Swarm Optimization applied to nuclear engineering problems
M. Waintraub, R. Schirru, C.M.N.A. Pereira
Progress in Nuclear Energy 51(2009) 680-688

[65] Swarm intelligence of artificial bees applied to in-core fuel management optimization
I.M.S. Oliveira, R. Schirru
Annals of Nuclear Energy 38(2011) 1039-1045

[66] Loading pattern optimization of PWR reactors using Artificial Bee Colony
O. Safarzadeh, A. Zolfaghari, A. Norouzi, H. Minuchehr
Annals of Nuclear Energy 38(2011) 2218-2226

[67] PWR loading pattern optimization using Harmony Search algorithm
N. Poursalehi, A. Zolfaghari, A. Minuchehr
Annals of Nuclear Energy 53(2013) 288-298

[68] Continuous firefly algorithm applied to PWR core pattern enhancement
N. Poursalehi, A. Zolfaghari, A. Minuchehr, H.K. Moghaddam
Nuclear Engineering and Design 258(2013) 107-115

[69] Mixed Integer Programming for Pressurized Water Reactor Fuel-Loading-Pattern Optimization
T.K. Kim, and C.H. Kim
Nuclear Science and Engineering 127(1997) 346-357

[70] Estimation of distribution algorithms for nuclear reactor fuel management optimization
S. Jiang, A.K. Ziver, J.N. Carter, C.C. Pain, A.J.H. Goddard, S. Franklin, H.J. Phillips
Annals of Nuclear Energy 33(2006) 1039-1057

[71] Two stochastic optimizationalgorithms applied to nuclear reactor core design
W.F. Sacco, R.E. Cassiano,C.M.N.A. Pereira
Progess in Nuclear Energy 48(2006) 525-539

[72] Nuclear reactor core optimization with Parallel Integer Coded Genetic Algorithm
A. Norouzi, M. Aghaie, A.R. MohamadiFard, A. Zolfaghari, A. Minuchehr
Annals of Nuclear Energy 60(2013) 308-315

[73] Comparison of metaheuristic optimization techniques for BWR fuel reloads pattern design
J.L. Francois, J.J. Ortiz-Servin, C. Martín-del-Campo, A. Castillo, J. E. Estrada,
Annals of Nuclear Energy 51(2013) 189-195

[74] Optimization of designing the core fuel loading pattern in a VVER-1000 nuclear power reactor using the genetic algorithm
M. RafieiKarahroudi, S.A. MousaviShirazi, K. Sepanloo
Annals of Nuclear Energy 57(2013) 142-150

[75] Nuclear fuel management optimization using genetic algorithms
M.D. DeChaine, M.A. Feltus
Nuclear Technology 111(1995) 109-114

[76] Performance evaluation of PSO and GA in PWR core loading pattern optimization
F. Khoshahval, H. Minuchehr, A. Zolfaghari
Nuclear Engineering and Design 241(2011) 799-808

[77] An enhanced integer coded genetic algorithm to optimize PWRs
A.Norouzi, A. Zolfaghari, A.H. Minuchehr, F. Khoshahval
Progress in Nuclear Energy 53(2011) 449-456

[78] A Quantitative Comparison of Loading PatternOptimization Methods forIn-Core Fuel Management of PWR
A. Yamamoto
Journal of Nuclear Science and Technology 34(1997) 339-347

[79] Application of genetic algorithms to optimize burnable poison placement in pressurized water reactors
S. Yilmaz, K. Ivanov, S. Levine, M. Mahgerefteh
Annals of Nuclear Energy 33(2006) 446-456

[80] Genetic Algorithm to Optimize the UO2/Gd2O3 Fuel Pin Designs in a Pressurized Water Reactor
S. Yilmaz, K. Ivanov, S. Levine, M. Mahgerefteh
Nuclear Technology 156(2006) 168-179

[81] Application of a hybrid method based on the combination of genetic algorithm and Hopfield neural network for burnable poison placement
F. Khoshahval, A. Fadaei
Annals of Nuclear Energy 47(2012) 62–68

[82] Fuel management optimization using genetic algorithms and expert knowledge
M.D. DeChaine, M.A. Feltus
Nuclear Science Engineering 124(1996) 188-196

[83] Study on the Optimization of Integral Fuel Burnable Absorbers Using the GeneticAlgorithm Based CIGARO Fuel ManagementSystem
B.V. Haibach, M.A. Feltus
Annals of Nuclear Energy 24(1997) 439-448

[84] A new approach to the use of genetic algorithmsto solve the pressurized water reactor's fuelmanagement optimization problem
J.L.C. Chapot, F. Carvalho Da Silva, R. Schirru
Annals of Nuclear Energy 26(1999) 641-655

[85] Pressurized water reactor reloading optimization using genetic algorithms
W. Hongchun
Annals of Nuclear Energy 28(2001) 1329-1341

[86] New genetic algorithms (GA) to optimize PWR reactors Part I: Loading pattern and burnable poison placement optimization techniques for PWRs
F. Alim, K. Ivanov, S.H. Levine
Annals of Nuclear Energy 35(2008) 93-112

[87] New genetic algorithms (GA) to optimize PWR reactors Part II: Simultaneous optimization of loading pattern and burnable poison placement for the TMI-1 reactor
F. Alim, K. Ivanov, S. Yilmaz, S.H. Levine
Annals of Nuclear Energy 35(2008) 113-120

[88] New genetic algorithms (GA) to optimize PWR reactors Part III: The Haling power depletion method for in-core fuel management analysis
F. Alim, K. Ivanov, S.H. Levine
Annals of Nuclear Energy 35(2008) 121-131

[89] Basic investigations related to genetic algorithms in core designs
C.M.N.A Pereira, R. Schirru, A.S. Martinez
Annals of Nuclear Energy 26(1999) 173-193

[90] The fuzzy clearing approach for a niching genetic algorithm applied to a nuclear reactor core design optimization problem
W.F. Sacco, M.D. Machado, C.M.N.A. Pereira, R. Schirru
Annals of Nuclear Energy 31(2004) 55-69

[91] AXIAL: a system for boiling water reactor fuel assembly axial optimization using genetic algorithms
C. Martin-del-Campo, J.L. Francois, H.A. Lopez
Annals of Nuclear Energy 28(2001) 1677-1682

[92] Advanced and flexible genetic algorithms for BWR fuel loading pattern optimization
C. Martin-del-Campo, Miguel-Angel, Palomera-Perez, J.L. Francois
Annals of Nuclear Energy 36(2009) 1553-1559

[93] An Order Coding Genetic Algorithm to Optimize Fuel Reloads in a Nuclear Boiling Water Reactor
J.J. Ortiz, I. Requena
Nuclear Science and Engineering 146(2004) 88-98

[94] A novel channel selection method for CANDU refuelling based on the BPANN and GA techniques
Xiaodong Huo, Zhongsheng Xie
Annals of Nuclear Energy 32(2005) 1081-1099

[95] Multicycle Optimization of Advanced Gas-Cooled Reactor Loading Patterns Using Genetic Algorithms
A. Kemal Ziver, J.N. Carter, C.C. Pain, C.R.E. Oliveira, A.J.H. Goddard, R.S. Overton
Nuclear Technology 141(2003) 122-141

[96] Genetic algorithms and artificial neural networks for loading pattern optimisation of advanced gas-cooled reactors
A.K. Ziver, C.C. Pain, J.N. Carter, C.R.E. Oliveira, A.J.H.Goddard, R.S. Overton
Annals of Nuclear Energy 31(2004) 431-457

[97] Application of a genetic algorithm to the fuel reload optimization for a research reactor
B. Quang Do, L.P. Nguyen
Applied Mathematics and Computation 187(2007) 977-988

[98] Application of Genetic algorithm in Research and Test Reactor Core Loading Pattern Optimization
P. Gang, P. Feng, F. Rong
PHYSOR 2002, Seoul, Korea, October 7-10 (2002)

[99] Multiobjective Pressurized Water Reactor Reload Core Design by Nondominated Genetic Algorithm Search
G.T. Parks
Nuclear Science and Engineering 124(1996) 178-187

[100] Evolutionary Multicriteria Optimization in Core Designs: Basic Investigations and Case Study
C.M.N.A. Pereira
Annals of Nuclear Energy 31(2004) 1251-1264

[101] Optimization of Boiling Water Reactor Loading Pattern Using an Improved Genetic Algorithm
Y. Kobayashi, E. Aiyoshi
Proceedings of the 2001 IEEE International Symposium on Intelligent Control Mexico, September 5-7 (2001)

[102] Optimization of Boiling Water Reactor Loading Pattern Using Two-Stage Genetic Algorithm
Y. Kobayashi, E. Aiyoshi
Nuclear Science and Engineering 142(2002) 119-139

[103] Optimization of a Boiling Water Reactor Loading Pattern Using an Improved Genetic Algorithm
Y. Kobayashi, E. Aiyoshi
Nuclear Technology 143(2013) 144-151

[104] An Evolutionary Optimization of the Refueling Simulation for a CANDU Reactor
B. Quang Do, H. Choi, G. Hong Roh
IEEE Transactions on Nuclear Science 53(2006)

[105] Multiobjective fuel management optimization for self-fuel-providing LMFBR using genetic algorithm
V.G. Toshinsky, H. Sekimoto, G.I.Toshinsky
Annals of Nuclear Energy 26(1999) 783-802

[106] A method to improve multiobjective genetic algorithm optimization of a self-fuel-providing LMFBR by niche induction among nondominated solutions
V.G. Toshinsky, H. Sekimoto, G.I. Toshinsky
Annals of Nuclear Energy 27(2000) 397-410

[107] Optimization of the core configuration design using a hybrid artificial intelligence algorithm for research reactors
A. Hedayat, H. Davilu, A.A. Barfrosh, K. Sepanloo
Nuclear Engineering and Design 239(2009) 2786-2799

[108] Nuclear reactor core optimization with Parallel Integer Coded Genetic Algorithm
A. Norouzi, M. Aghaie, A.R.Mohamadi Fard, A. Zolfaghari, A. Minuchehr
Annals of Nuclear Energy 60(2013) 308-315

[109] Application of the Distributed Genetic Algorithm for In-Core Fuel Optimization Problems under Parallel Computational Environment
A. Yamamoto, H. Hashimoto
Journal of Nuclear Science and Technology, 39(2002) 1281-1288

[110] Coarse-grained parallel genetic algorithm applied to a nuclear reactor core design optimization problem
C.M.N.A. Pereira, C.M.F. Lapa
Annals *of Nuclear Energy* 30(2003) 555-565

[111] A parallel genetic algorithm with niching technique applied to a nuclear reactor core design optimization problem
C.M.N.A. Pereira, W.F. Sacco
Progress in Nuclear Energy 50(2008) 740-746

[112] Genetic Algorithms in Search Optimization and Machine Learning
D.E. Goldberg
Addison-Wesley, Reading, MA, (1989)

[113] Genetic Algorithm + Data Structure = Evolution Programs
Z. Michalewicz
Spinger-Verlag New York, (1994)

[114] A comparative study of some real-coded genetic algorithms for unconstrained global optimization
B.A. Sawyerr, M.M. Ali, A.O. Adewumi
Optimization Methods and Software 6(2011) 945-970

[115] Real-coded genetic algorithm with uniform random local search
B.A. Sawyerr, M.M. Ali, A.O. Adewumi
Applied Mathematics and Computation 228(2014) 589-597

[116] Multi-Objective Optimization using Evolutionary Algorithms
K. Deb
John Wiley & Sons Ltd (2001)

[117] Adapting Crossover and Mutation Rates in Genetic Algorithms
W.Y. Lin, W.Y. Lee, Tzung-Pei Hong
Journal of Information Science and Engineering 19(2003) 889-903.

[118] Foundations of Genetic Algorithms
D.E. Goldberg, K. Deb, Edited by Gregory J.E. Rawlins
Morgan Kaufmann Publishers, Inc. (1991)

[119] Multiobjective Genetic Algorithms for Clustering
U. Maulik, S. Bandyopadhyay, A. Mukhopadhyay
Spinger-Verlag, Berlin Heidelberg (2011) 36-43

[120] Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms
J.D. Schaffer
Ph. D Thesis, Vanderbilt University, Nashville, TN (1984)

[121] Genetic search strategies in multi-criterion optimal design
P. Haleja, C.Y. Lin
Structural Optimization 2(1992) 99-107

[122] Genetic algorithms for multiobjective optimization: Formulation, discussion, and generalization
C.M. Fonseca, P.J. Fleming
Proceedings of Fifth International Conference on Genetic Algorithms (1993) 416-423

[123] Multiobjective optimization using nondominated sorting in genetic algorithms
N. Srinivas, K. Deb
Evolutionary Computation 3(1994) 221-248

[124] A niched pareto genetic algorithm for multiobjective optimization
J.Horn, N. Nafpliotis, D.E. Goldberg
Proceedings of the First IEEE Conference on Evolutionary Computation,
IEEE World Congress on Computational Intelligence 1(1994) 82-87

[125] A new method to solve generalized multicriteria optimization problems using simple genetic algorithm
A. Osyczka, S. Kundu
Structural Optimization 10(1995) 94-99

[126] An evolutionary algorithm for multiobjective optimization:The strength Pareto approach
E. Zitzler, L. Thiele
Technical Report, Switzerland Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology
Zurich 43(1998)

[127] Approximating the non-dominated front using the Pareto archived evolution strategy
J.D. Knowles, D.W. Corne
Evolutionary Computation Journal 2(2000) 149-172

[128] A Fast and Elitist Multiobjective Genetic Algorithm:NSGA-II
K. Deb, A. Pratap, S. Agarwal, T. Meyarivan
IEEE Transactions on Evolutionary Computation 6(2002) 182-197

[129] Multi-objective optimization using genetic algorithms: a tutorial
A. Konak, D.W. Coit, A.E. Smith
Reliability Engineering Systems Safety 91(2006) 992-1007

[130] http://www.igcar.ernet.in
Indira Gandhi Cetre of Atomic Research (IGCAR), (accessed 03-July-2015)

[131] The design of the Prototype Fast Breeder Reactor
S.C. Chetal, V. Balasubramaniyan, P. Chellapandi, P. Mohanakrishnan,
P. Puthiyavinayagam, C.P. Pillai, S. Raghupathy, T.K. Shanmugham, C. S.Pillai
Nuclear Engineering and Design 236(2006), 852-860

[132] Optimisation of Steam Surface Condenser
A. Sen
Internal Design Note for PFBR by Indira Gandhi Centre for Atomic Research (IGCAR, India)
Doc. No. PFBR/71200/DN/2053 Rev.4(2007)

[133] Heat Sink Optimization in Power Plants using Mathematical Programming Methods
S.K. Hati, R. Bain
R&D Centre, DCL, India, (1995)

[134] Heat Exchange Institute Standards for Steam Surface Condensers
HEI, 10th Edition, (2006)

[135] http://www.npcil.nic.in
Nuclear Power Corporation India Ltd, (accessed 03-July-2015)

[136] The Indian PHWR
S.S. Bajaj, A.R. Gore
Nuclear Engineering and Design 236(2006), 701-722

[137] Reactor physics computer code development for neutronic design, fuel-management, reactor operation, and safety analysis of PHWRs
B.P. Rastogi
International Atomic Energy Agency (IAEA)
https://www.iaea.org/inis/ (1989)

[138] Computer codes for fuel management of PHWRs
K.R. Srinivasan
TAQUIL and TRIVENI Internal report of Bhabha Atomic Research Centre, Bombay (India)
Note No. PHWR/500/PHY/18(1986)

[139] Opening the Box of a Boxplot
Y. Benjamini
The American Statistician 42(1988) 257-262

[140] http://www.bhavini.nic.in
Bharatiya Nabhikiya Vidyut Nigam Ltd, (accessed 03-July-2015)

[141] Indian fast reactor technology: Current status and future programme
S.C. Chetal, P. Chellapandi
Sadhana, Indian Academy of Sciences 38(2013) 795-815

[142] Current status of fast reactors and future plans in India
S.C. Chetal, P. Chellapandi, P. Puthiyavinayagam, S. Raghupathy,
V. Balasubramaniyan, P. Selvaraj, P. Mohanakrishnan, Beldev Raj
Energy Procedia, 7(2011) 64-73

[143] Approach to equilibrium fuelling scheme of 500 MWe PFBR based 3-D core burnup modeling
K. Devan, A. Riyas, P. Mohanakrishnan
Nuclear Engineering and Design 241(2011) 1596-1605

[144] http://www.r-project.org/
R Project, (2015), (accessed 03-July-2015)

[145] Applied Meta-Analysis with R
D.G. Chen, K.E. Peace
CRC Press, Taylor & Francis Group (2013)

[146] ABBN-93 Group Data Library Part I: Nuclear Data for the Calculation of Neutron and Photon Radiation Functions INDC (CCP-409)
G.N. Manturov
IAEA, Vienna, (1997)

[147] An Interface between CONSYST/ABBN-93 and IGCAR Diffusion Theory Codes
K. Devan
Internal Report by Indira Gandhi Centre for Atomic Research (IGCAR, India)
Report No.RPD/NDS/90 (2003)

[148] Studies on physics parameters of metal (U-Pu-Zr) fuelled FBR cores
A. Riyas, P. Mohanakrishnan
Annals of Nuclear Energy 35(2008) 87-92

[149] ULOF transient behaviour of metal-fuelled fast breeder reactor cores as a function of core size and perturbation methods
Riyas, P. Mohanakrishnan
Nuclear Engineering and Design 278(2014) 141-149.

[150] Inherent safety aspects of metal fuelled FBR
T. Sathiyasheela, A. Riyas, R. Sukanya, P. Mohanakrishnan, S.C. Chetal
Nuclear Engineering and Design 265(2013) 1149-1158