AN ITERATIVE OPTIMIZATION METHOD USING GENETIC ALGORITHMS AND GAUSSIAN PROCESS BASED REGRESSION IN NUCLEAR REACTOR DESIGN APPLICATIONS

A Dissertation

by

AKANSHA KUMAR

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2016

Major Subject: Nuclear Engineering

ABSTRACT

The optimization of a complex system involves the determination of optimum values for a set of design parameters. The optimization search happens in order to meet a specific set of objectives concerning the quantities of interest (QOI). Also, the design parameters are a subset of the input parameters and the QOIs are determined from the output parameters. Particularly, when the parameter space is large, optimization necessitates a significant number of executions of the simulator to obtain a desired solution in tolerance limits. When the simulations are expensive in terms of computation time, an emulator based on regression methods is useful for predictions. This work presents a novel methodology that uses an iterative hybrid global optimization method (GOM) using genetic algorithms (GA) and simulated annealing (SA) model coupled (HYBGASA) with a Gaussian process regression method based emulator (GPMEM) to optimize a set of input parameters based on a set of defined objectives in a nuclear reactor power system. Hereafter this iterative hybrid method comprising of HYBGASA and GPMEM would be called as the "IHGOM". In addition to optimization, IHGOM iteratively updates the trial data obtained from the neighborhood of the near optimal solution, used to train the GPMEM in order to reduce regression errors. The objective is to develop, model and analyze IHGOM, and apply it to an optimization problem in the design of a nuclear reactor. Development and analysis of IHGOM and its implementation in a nuclear reactor power system problem is a significant contribution to the optimization and the nuclear engineering communities.

DEDICATION

This dissertation is dedicated to the unwavering love, affection, motivation, encouragement, and support of my parents, Dr. A.G Patrudu and Dr. Ajita Patra, my wife Vyshali, and my sister Anisha Anusaranya.

ACKNOWLEDGEMENTS

I would like to express my sincerest appreciation to my advisor Dr. Tsvetkov for his support, advice, guidance, and mentoring in conducting the research and finishing this dissertation. I would also like to thank my committee members (Dr. Adams, Dr. Ragusa, Dr. Petrova and Dr. McClarren) for their guidance over the past few years towards this dissertation. I am also grateful to my colleagues in my research group as well as in my department for their selfless support during times of need.

NOMENCLATURE

GA	Genetic Algorithms
SA	Simulated Annealing
GP	Gaussian Processes
GOM	Global Optimization Method
RA	Regression Analysis
HYBGASA	Hybrid Genetic Algorithms and Simulated Annealing
LO	Local Optimization
POS	Pareto Optimal Solution
GCFBR	Gas Cooled Fast Breeder Reactor
GPMEM	Gaussian Processes Based Emulator
IHGOM	Iterative Hybrid Global Optimization Method
GPR	Gaussian Processes Based Regression

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

This chapter presents the dissertation outline, background and motivation driving the work. An introduction is presented for optimization and regression, along with the objectives. An overview is given of existing work related to optimization in nuclear engineering, applications of regression, and the application of hybrid methods in optimization.

1.1 Dissertation Outline

The dissertation begins with an introduction to the developed optimization method. In this chapter, the background and motivation in the development of the optimization method is presented. A detailed description of the objectives and tasks in the development of the method is given. This chapter includes an introduction to the theoretical aspects of optimization and regression. Overall, this chapter presents an outline and a foundation on which this research has been built.

Following the introduction, the description of the novel method developed in this research is presented in Chapter 2. This chapter describes all the necessary modules and presents a graphical solution flow of the work. The implementation details and convergence criteria is presented in this chapter. This chapter presents a detailed view of the selection of the tuning/hyper-parameters and sensitivity studies.

Following the method description, an implementation on a set of problems demonstrating the novel method is presented in Chapter 3. The method is implemented to solve a test problem and two defined reactor problems. A detailed problem description with analysis of the results are presented in this chapter.

The dissertation is concluded with a recap of the objectives and a summary. A discussion on prospective applications and paths for future work are presented that includes an intense sensitivity study and method improvements.

1.2 Background and Motivation

The optimization of a complex system involves the determination of optimum values for a set of design parameters in order to meet a specific set of objectives based concerning the quantities of interest (QOI) in which the design parameters are a subset of the input parameters and the QOIs are determined from the output parameters. The system can be an experiment or a computational model. Particularly, when the parameter space is large, optimization necessitates a significant number of executions of the system to obtain a desired solution in tolerance limits. When the simulations are expensive in terms of computation time, an emulator may be used. The emulator is based on regression methods and is used as a black box to predict the values for the output parameters based on the values of the input parameters. The motivation for this study is to develop an optimization method that has an efficient search scheme and is fast enough to reach a desired optimal solution.

1.2.1 Optimization in Nuclear Engineering

Previous work related to optimization in nuclear engineering that uses GA includes core design [1, 2, 3], plant design [4], nuclear system availability and maintenance scheduling [5], fuel management [6] and spent fuel management [7]. However, coupled neutronics-thermal hydraulics problems have not been explored, and the effectiveness of using GA in solving coupled problems have not been evaluated.

1.2.2 Hybrid Optimization Methods

Researchers have explored non hybrid GOM based on Gaussian processes, but those are limited to the concept of expected improvement [8, 9], and did not involve a global evolution-based optimization strategy. GA has had the problem of having local convergence of parameters, which forces the search to explore other regions using a SA based hybrid [10, 11, 12, 13] method. These articles show the hybrid method has performed better than other non-hybrid GOMs. SA replaces the traditional “1-flip neighborhood” local optimization method in GA to take care of the issue of local convergence. Even though GOM implementation is dependent on the problem, a common iterative strategy is always used in all the multi-objective problems.

1.2.3 Regression

Predictive analysis using regression, response surfaces, Gaussian processes, and Bayesian methods have had significant acceptance in the research community. Particularly, machine learning using GPMEM is extensively used wherein several complex system have been emulated [14, 15, 16, 17, 18] using Gaussian process based regression methods. However, very limited work is performed in the nuclear reactor design domain. Therefore, an integrated approach of a global evolution based optimization method and Gaussian processes applied to a coupled neutronics-thermal hydraulics problem would be a significant contribution to the nuclear engineering research community.

1.3 Global Optimization Methods

This section presents an overview of the theoretical background of optimization, types of optimization methods, and heuristic methods.

1.3.1 Introduction to Optimization

Optimization is the process of the determination of a set of values for the design parameters that solves a maximization or minimization function of a set of objectives derived from the QOIs. The QOIs are functions of the design parameters, but the functional forms are not known. Let S be a set of feasible solutions of the design parameters, and f is a fitness function such that:

$$f : S \rightarrow \mathbb{R}, \tag{1.1}$$

and f is a quantifiable form derived from the objective functions. The goal is to determine a globally optimal solution i.e. find a feasible optimal solution $s^* \in S$ such that,

$$f\left(s^*\right) \leq f\left(s\right), \text{for all}\, s \in S. \tag{1.2}$$

Let

$$f^* = f\left(s^*\right) = \min_{s \in S} f\left(s\right), \tag{1.3}$$

is the optimal cost for a minimization function. Therefore,

$$S^* = \{s \in S : f\left(s\right) = f^*\} = arg\, \min_{s \in S} f\left(s\right), \tag{1.4}$$

defines the set of pareto optimal solutions. If optimization is performed in the global design space it is called a GOM. When the minimization function is a combination of several minimization/maximization functions, it is called a multi-objective optimization search (MOOS). In MOOS, the objectives can be conflicting. This means that there are trade-off solutions and therefore there is no single optimum solution, but rather a number of pareto optimal solutions. An inherent property of the pareto optimal solutions is that no solution from the pareto set can be said to be better than any other.

1.3.2 Neighborhood

Neighborhood is defined as $N(s)$ where $N(s) \subset S$ and $s \in S$ are a set of solutions close to s. Each $i \in N(s)$ is a neighbor of s. For an instance (S, f) of a combinatorial optimization problem and a neighborhood N, a solution $\hat{s}$ is locally minimal with respect to N if:

$$f(\hat{s}) \leq f(i), \text{for all}\, i \in N(\hat{s}), \tag{1.5}$$

with $\hat{S}$ denoting the set set of local optimal solutions of (S, f). Any local solution is global optimal if in the neighborhood N defined for a problem P, and for any instance (S, f) of P, $\hat{S} \subseteq S^*$.

1.3.3 Optimization Methods

Optimization methods are broadly divided into three categories:

1.3.3.1 Exact Algorithms

Exact algorithms are guaranteed to find a global optimal solution. In most cases where there is a significant number of design parameters are close to NP hard, the run times to obtain the optimal solution are unreasonably high. The effort usually grows polynomially with problem size for most of the P problems, however for NP problems the effort grows exponentially with problem size. A subset of popular exact algorithms are the simplex method, branch-and-bound methods, and Benders decomposition.

1.3.3.2 Approximate Algorithms

Approximate algorithms are guaranteed to find an approximate solution within a known approximation ratio. These algorithms are relatively faster than exact algorithms for large-scale problem instances. For many hard problems, it can be shown that, unless some very unlikely statement is true, there cannot be an efficient approximation algorithm with a constant approximation ratio.

1.3.3.3 Heuristic Algorithms

Heuristic algorithms find a "good" solution "fast". There is no guarantee on the quality of the solution in general. Optimization of large instances with a large parameter space is difficult using exact algorithms and therefore the best choice is heuristic methods. Heuristic methods usually imitate a natural process such as annealing, biological evolution, food foraging, and bird flocking. Popular heuristic methods are tabu search (TS), evolutionary algorithms (EA), ant colony optimization, simulated annealing (SA), and particle swarm optimization (PSO). The focus of current research is heuristics algorithms.

1.3.4 Heuristic Methods

The term heuristics, stems from the Greek word *heuriskein*, which means to find or discover. Heuristics methods are used for search, discovery, learning, and problem-solving of large-scale practical problems. The solutions are not guaranteed to be perfect, but are acceptably good solutions with relative speed. Heuristic methods are based on the following generic execution steps:

1.3.4.1 Construction Heuristics

Start with a feasible solution from scratch. This solution can be randomly generated or else generated from predefined conditions. Sometimes greedy methods are used to generate the construction heuristics.

1.3.4.2 Local Search

Given a feasible solution, iteratively search for a better neighbor until a local optimum is obtained. Two common methods in local search are “best improvement” and “first improvement”. In “best improvement”, the neighborhood is extensively searched and the best local optimum is obtained. However, in “first improvement”, the first fitter solution in the neighborhood is used as the local optimum. These methods are dependent on the problem.

1.3.4.3 Meta-heuristics

Once the local optimum is obtained, a search beyond the neighborhood is performed. Usually randomized methods are used to escape poor quality local optimum, and the search is converged to a global optimum. To increase the speed and the quality of the solution, several meta-heuristic strategies are combined together and hybrid algorithms are employed.

A brief introduction to popular heuristics methods are presented as follows:

1.3.4.3.1 Tabu Search: Tabu search [19, 20, 21] is a local search method that uses memory to store information about previously visited solutions, and hence restricts future searches to non visited solutions only. It is very effective in the sense that it avoids repetitive searches, but it is also memory intensive. This method is usually used with other local search methods to guide those to escape from local optimality traps. Following steps describe the Tabu search method:

- Start with an initial random solution.
- Create a candidate list of moves where these moves are the prospective new solutions.

- From the candidate list determine the best admissible candidate based on tabu restrictions and aspiration criteria. Designate this solution based on the best admissible candidate as the new current solution and record it as the new best solution if it improves the previous best solution.

- Repeat the above steps until a defined stopping criteria is obtained. Update the aspiration criteria and tabu restrictions if desired.

1.3.4.3.2 Genetic Algorithms: GA is a search heuristic machine learning model which is derived from the process of natural selection based on the theory of species evolution [22]. It involves the processes behind natural selection, such as inheritance, reproduction, crossover, mutation. A detailed description of GA is presented in Section:1.3.4.4.

1.3.4.3.3 Particle Swarm Optimization: Particle Swarm Optimization (PSO) [23, 24] is a population based stochastic optimization method developed by Dr. Eberhart and Dr. Kennedy [25] in 1995, inspired by the social behavior of bird flocking or fish schooling. It is an evolution based method where a particle keeps track of its coordinates in the problem space which are associated with the best solution it has achieved so far. The method starts with an initial random solution and searches for a pareto optimal solution by updating generations. GA and PSO have lots of similarities, however, PSO does not have operations such as mutation and crossover. Instead, each particle keeps track of the change in velocity towards the global and local best solution. The following simple analogy related to bird flocking describes the PSO method . Suppose a group of birds are randomly searching for food in an area and there is only one piece of food in the area being searched. At the beginning the birds do not know where the food is, but they know how far the food is in each iteration. How do they know that? They follow the bird which is nearest to the food. PSO is a continuous technique, therefore it is not suitable for combinatorial problems.

1.3.4.3.4 Ant Colony Optimization: Ant Colony Optimization (ACO) [26, 20] is a search method used to determine the optimal path in a graph based on the foraging behavior of ants. Ants, while seeking a path between their colony and a source of food, deposit pheromones on the ground in order to mark a favorable path for other members on the colony to follow. Other ants who follow perceive the presence of pheromones and tend to follow paths where the pheromone concentration is higher. In this way the ants iteratively determine an effective way to obtain food. The optimization method, ACO uses this foraging mechanism to determine an optimal solution. The following steps describe the ACO method:

- Initialize pheromone trails.
- While the stopping criteria is not met
    - Construct ant solutions
    - Apply local search
    - Update pheromone values

1.3.4.3.5 Greedy Randomized Adaptive Search Procedures: Greedy Randomized Adaptive Search Procedures (GRASP) is a greedy adaptive search method used to determine a global optimal solution. In GRASP, each iteration consists of the construction of a greedy randomized feasible solution followed by finding a local optimal solution in the neighborhood of the feasible solution. In one of the implementations of GRASP, an elite solution obtained randomly is used as a guiding solution and then GRASP is used to link a path from the initial solution to the guiding solution. It is an effective search method, however it is very expensive due to its greedy nature.

1.3.4.4 Genetic Algorithms

This section starts with a biological background of GA. Every living organism consists of animal cells with every cell consisting of a nucleus that has the genetic information of an organism. The DNA molecule in the nucleus consists of thread-like structures called chromosomes. The chromosome is a constitution of genes with a gene located at specific locations called locus. A gene is the basic physical and functional unit consisting of instructions that define an organism i.e. how the organism survives, how it appears, and how it behaves in its environment. These characteristics determine the adaptability of the organism in the environment, referred to as the fitness of the organism. Basically, a gene encodes a trait of the organism, e.g. color of the skin.

In GA, a population of individuals i.e. a set of chromosomes to an optimization problem, is manipulated using the above mentioned processes to evolve towards a new generation of population with stronger individuals. The chromosome consists of a set of genes that carry intrinsic characteristics of a symbolic individual. The adaptation capability known as the fitness of an individual in the environment depends on these intrinsic characteristics. In GA [27, 28], the selection and evolution process is defined in such a way that only the stronger individuals, i.e. the individuals having a higher fitness level, in a generation pass their characteristics to their off-spring, hence making them stronger. Therefore, the population in a newer generation is more fit as compared to the population in the previous generation.

The GA flow starts with a random set of individuals selected from a set of possible configurations i.e. a set of possible values for the input parameters. These are referred to as a “population” in a “generation”, with the first set of individuals called the “initial” population or the “first” generation. Each individual is then evaluated, and a “fitness” value for that individual is calculated. This is the stage where the modules are executed and a solution is obtained for the input parameters given by the “individual”. The fitness value is calculated by how well the solution fits to our objectives. This stage is called the “evaluation” stage. Next, we select a set of fit individuals from the population to obtain a “new” population for the next generation. Selection is made such that “bad” designs (individuals with low fitness value) are discarded and “good” designs are carried forward to the next generation. The selected individuals are called “parents”. This stage is called the “selection” stage, and the set of selected individuals form a “mating pool”. Then, crossover is performed by creating crosses of the parents i.e. the individuals in the “mating pool”, to create a set of even “fitter” individuals. The idea is that the individuals of the new population inherit the best characteristics of their parents. This stage is called as the “crossover” stage. Then, we perform the evaluation stage using this new population, and the above steps are performed iteratively until the desired fitness is obtained. Fig: 1.1 presents a graphical view of GA.

The advantage of GA over other non-population based optimization methods is that GAs work with a population of solutions instead of a single solution. Therefore, more than one string is processed simultaneously and used to update other strings in the population. GAs do not necessarily require any additional information such as the gradient to help in the search directions, which makes it simple and intuitive. GAs use probabilistic rules in the search and it sees the system which provides the fitness value as a black-box. In classical methods, where there is a coupling between the underlying physics and the search method, there is coping for transition rules, which assist in the search directions the methods that are not robust. This is unlike GAs in which the stochasticity and the absence of transition rules make it more effective and widely usable. Another advantage of Gas is that it is highly scalable and

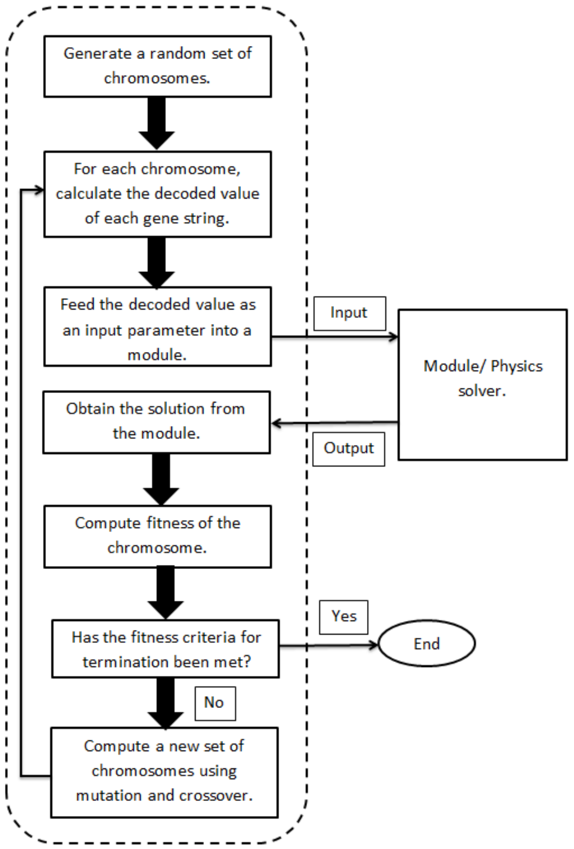

Generate a random set of chromosomes.
For each chromosome, calculate the decoded value of each gene string.
Feed the decoded value as an input parameter into a module.
Input
Module/ Physics solver.
Obtain the solution from the module.
Output
Compute fitness of the chromosome.
Has the fitness criteria for termination been met?
Yes
End
No
Compute a new set of chromosomes using mutation and crossover.

Figure 1.1: Genetic algorithms flow chart

parallel executable. In a population, the chromosomes can be executed in parallel, the crossover, mutation and selection operators can be executed in parallel.

1.3.4.4.1 Binary GA: The following illustration presents the implementation details of GA, and discus the operations, crossover, mutation, and fitness calculation. Let us assume two types of input parameters given by G_1, and G_2. Each of these input parameters are a gene, and a combination of these form a chromosome. For the genes with a size of four bits, the size of the chromosome is eight bits. Let C_{11}, and C_{12} be two chromosomes selected randomly from the mating pool after the selection stage:

$$C_{11} = \underline{01101100} \tag{1.6}$$

$$C_{12} = \overline{11100010} \tag{1.7}$$

Suppose the crossover point is defined as the fifth bit. Then the bits after the fifth bit in the parent-chromosomes, C_{11}, and C_{12} are swapped to create two children-chromosomes, C_{21}, and C_{22}, given by:

$$C_{21} = \underline{01101}\overline{010} \tag{1.8}$$

$$C_{22} = \overline{11100}\underline{100} \tag{1.9}$$

The chromosomes, C_{21}, and C_{22}, retain some characteristics of their parent-chromosomes, C_{11}, and C_{12}, and those will explore the solution space not explored by the parents, C_{11}, and C_{12}. This operation is called as the "crossover" operation. The mutation operation is performed, when a specific bit in the chromosome is changed. Suppose a chromosome C_{30} is defined as:

$$C_{30} = 01101010 \tag{1.10}$$

has the mutation operation on its fourth bit, resulting in:

$$C_{33} = 011\underline{1}1010 \tag{1.11}$$

where the bit 0 at the fourth position in C_{30} changes to 1 in C_{33}. Mutation helps in exploring different regions of the search space and prevents from stagnation. Mutation plays an important role in species diversity. Mutation gives the children-chromosome characteristics that may be very unlike those of its parents-chromosomes, increasing the overall diversity of the population, and therefore enhancing exploration of the search space.

To quantify how close a solution is to the specified objectives, a value is calculated and assigned to each chromosome after it is evaluated and a solution is obtained. This value is called a "fitness" value. In other words, the fitness of a chromosome is a function of the variables that form the objective. These variables can have the objective of maximization, minimization, or proximity to limits. Let us define two variables, O_1 and O_2. Suppose the objectives are to maximize O_1, and minimize O_2. Let ΔO_1 and ΔO_2 define the distance of the variables O_1 and O_2 from the constraint limits. The fitness function is defined as:

$$f = \frac{1}{\Delta O_1} + \frac{1}{\Delta O_2} \tag{1.12}$$

The fitness function is problem dependent and custom functions are built based on the design parameters and objectives. The general scheme of the steps involved in the traditional GA [20] based GOM is presented as follows:

1. Generate an initial population using LHS, $S = s_i, s_2, s_3 ... s_N$, where s_i is a chromosome, and N is the population size

2. Apply a local search algorithm A_{local} to each chromosome, s_i, and replace each s_i with its local optimum

3. While terminating criteria is not satisfied:

   3.1. Select $\hat{K} = \hat{k}_1, \hat{k}_2, \hat{k}_3, ... \hat{k}_M$ distinct subsets of size *two* as parents.

   3.2. For each $\hat{k}_i$, perform a **crossover** operation. This gives rise to *two* new solutions (children).

3.3. Apply local search algorithm A_{local} to each of the $\hat{K}$ new solutions resulting in set $\hat{S}$ of solutions.

3.4. Choose N survivors from $S \cup \hat{S}$ using a **selection** strategy.

3.5. If required perform a **mutation** operation.

4. Return the converged population S, and the best s_i.

1.3.4.5 Simulated Annealing

Simulated Annealing (SA) [29] is one of the most effective threshold algorithms used for iterative improvement of search in optimization. In SA, instead of rejecting non-useful solutions, these are accepted at a certain probability. Therefore the algorithms elect to keep a non-useful solution, and hence is able to escape local maxima/minima and avoid sub-optimal convergence. SA by itself is a meta-heuristic optimization method but is not suitable for large search spaces. The name, SA comes from annealing in metallurgy, which involves heating and cooling of a material to increase the size of its crystals and reduce defects. SA is an adaptation of the Metropolis-Hastings algorithm to generate sample states of a thermodynamic system. In condensed matter physics, annealing is a thermal process for obtaining low-energy states of a solid in a heat bath. SA is implemented using the following basic steps:

1. Start a random initial solution (s_{old}) for a maximization problem.

2. Calculate its cost ($f\left(s_{old}\right)$)

3. Generate a random neighboring solution s_{new} with a cost ($f\left(s_{new}\right)$)

4. Compare s_{old} with s_{new},

4.1. if $f\left(s_{new}\right) \geq f\left(s_{old}\right)$, move to the new solution and discard s_{old}

4.2. if $f(s_{new}) < f(s_{old})$, "may be" move to a new solution based on a probability,

$$P_c(s_{new}) = \exp \frac{f(s_{new}) - f(s_{old})}{c}, \tag{1.13}$$

where $P_c(s_{new})$ is the probability of acceptance of solution s_{new}, and c is a control parameter.

5. Repeat steps 3 and 4 until an acceptable optimal solution is found based on some convergence criteria.

The control parameter is analogous to the "temperature parameter" in annealing, and the method for choosing c is called the "cooling schedule". For large values of c, large increases in cost are accepted with high probability. As c decreases, only smaller increases are accepted. As c approaches 0, no increases are accepted at all. It means that the algorithm is more likely to accept sort-of-bad jumps than really-bad jumps, and is more likely to accept them early on, when the temperature is high.

1.4 Regression

This section presents a detailed description of regression, predictive analysis using regression, Gaussian processes based regression (GPR) and model prediction error (MPE).

1.4.1 Introduction to Regression

Regression is a statistical estimate of the relationship between parameters based on the observations in a practical problem. In particular, when the functional form is not known, regression builds a surrogate model and helps to determine how a dependent variable would be affected due to a change in the independent variables. This surrogate model used as a black box as an effective tool for predictive analysis. Regression includes several techniques that define a relationship between a dependent variable and one or more independent variables. The simplest form of this relationship can be thought of as the equation of a line, $y = mx + c$, where the slope m and the y-intercept c are determined from a set of two-dimensional observations (x, y). Now with a known m and c a prediction for y^* for any x^* van be reliably done at a confidence level. Interpolation, extrapolation, least squares fit, expected improvement, and spline interpolation are a few standard methods used for predictive modeling. Other regression methods that are much finer than those mentioned above are Bayesian methods, Multivariate Adaptive Regression Splines (MARS), Markhov chain Monte Carlo (MCMC) and GPR. In regression, due to the fact that the surrogate model is obtained from observations, there is an inherent regression error induced into the system. Intuitively, for an under-predicted model, the more observations, the less the magnitude of the regression error. The error in predictions i.e. the deviation of the predictions from the actual value due to regression errors is called as the MPE.

1.4.2 Need For Regression

Regression uses only the observations to build a surrogate model which is used for predictions. Apparently, the surrogate model does not depend on the source of the observations and therefore it does not care about the physical phenomenon that determined the values for the dependent parameters. Therefore, it can be used as a black box and it is significantly fast as compared to the actual physical model. A standard process used in regression is: firstly use a training set obtained from the observations to train/supervise the model and build the regression fit, and then predict the values for the dependent variables based on the test input data.

1.4.3 Gaussian Process Based Regression

In this section a detailed analysis of GPR method with the governing equations are presented. The complex system can be represented in a simple symbolic functional for:

$$Y = f(X), \tag{1.14}$$

where X given by:

$$X = [x_1, x_2, ..., x_K], \tag{1.15}$$

is a vector of K input parameters, and Y given by:

$$Y = [y_1, y_2, ..., y_L], \tag{1.16}$$

is a vector of L output parameters. Each output parameter, y_i is a function of X, given by

$$y_i = F(X) + \epsilon, \tag{1.17}$$

where ϵ is a random noise in the observations. In a system where the functional form (1.14) is not known, and only a set of trial data from known observation is given, regression models such as GP are used to emulate the functional form and perform predictions for a set of input test data. The input (X) in the nuclear power system domain can be thought as of variables such as enrichment, moderator ratio, etc., and the output (Y) parameters are the QOIs such as flux, peaking factors, thermal efficiency etc. Gaussian process based regression is a powerful tool to emulate the functional form (1.14).

1.4.3.1 Introduction to Gaussian Processes

A Gaussian process (GP) is a statistical distribution of data throughout some domain for which any finite linear combination of samples follows a multivariate Gaussian distribution. The N observations in an arbitrary data set can be imagined as a single point sampled from some multivariate Gaussian distribution. Therefore, working backwards, this Gaussian distribution can provide information about the predictions. Figure.1.2 has some data points (red dots) for independent variable x.

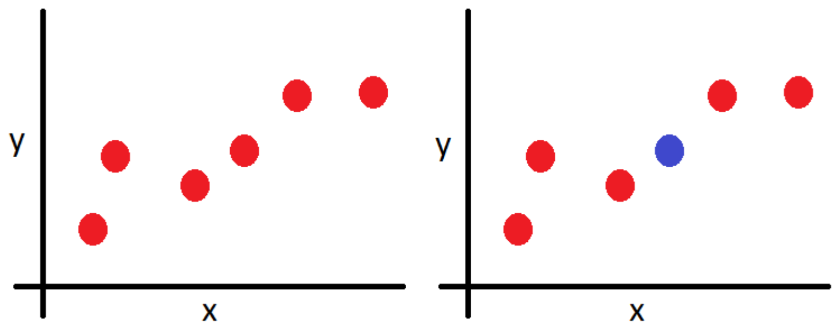


Figure 1.2: Random function with the functional form not known

The function $f(x)$ is not known and the likelihood of the point (blue dot) in the function is not known. GP is used to get an estimate of the source of these points in a confidence level and gives a best estimate of the dependent variable at some new value x^* (the blue dot in Figure.1.2). If $f(x)$ is known as linear, then with some assumptions, a least-squares method could be used to fit the data. Similarly if $f(x)$ is known to be quadratic, cubic, or any other known form, then standard principles of model selection could be used as the prediction model. But in the case where there are no clues about the functional form, GPR comes to picture. Instead of relating $f(x)$ to any known form, a GP is used to represent it rigorously by allowing observed data to supervise the learning of the model. With some assumptions, GP helps in answering the following questions:

- There are some data points, how to rank the likelihood of the functions ?
- What is the expected function i.e. where the function will most likely be ?
- The function might look like any of a set of example functions by sampling from the posterior distribution.
- Here is a prediction of what the function will evaluate at the test data at a confidence level.

1.4.3.2 Gaussian Process Based Emulator

The surrogate model built using GPR is called the Gaussian process based emulator (GPMEM)[30]. An observation i related to an underlying function (1.17) with a Gaussian noise is given by:

$$y_i = f(X) + \mathcal{N}\left(0, \sigma_n^2\right), \tag{1.18}$$

where $\mathcal{N}\left(0, \sigma_n^2\right)$ is the Gaussian noise in the observation with a standard deviation of σ_n. For simplicity, the suffix i in y_i and x_i are dropped. In other words Y is assumed to have a single parameter y and X has a single parameter x. GP is used with a basic assumption that x can be represented as a sample from a multivariate Gaussian distribution, therefore the distribution to prior data over the parameters is given by:

$$(y) \sim \mathcal{N}\left(0, K\left(x, x\right)\right), \tag{1.19}$$

where K is the kernel function that creates the covariance matrix, and $\mathcal{N}$ is the normal distribution. A squared exponential form is used for the covariance function to generate the covariance matrix. This covariance function relates one observation to another. For any two observations, $\hat{x}$ and $\bar{x}$, the covariance function, $K\left(\hat{x}, \bar{x}\right)$ is given by:

$$k\left(\hat{x}, \bar{x}\right) = \sigma_f^2 \exp \frac{-\left(\hat{x} - \bar{x}\right)^2}{2l^2}, \tag{1.20}$$

where σ_f is the output variance, and l is the length parameter. σ_f is a scale factor that determines the average distance of the function away from its mean. l acts as a flexibility parameters that controls the "wiggles" in the function. In other words, extrapolation cannot be trusted l units away from the data. The noise presented in (1.18) propagates to the covariance function (1.20) and yields:

$$k\left(\hat{x}, \bar{x}\right) = \sigma_f^2 \exp \frac{-\left(\hat{x} - \bar{x}\right)^2}{2l^2} + \sigma_n^2 \delta\left(\hat{x}, \bar{x}\right), \tag{1.21}$$

where $\delta\left(\hat{x}, \bar{x}\right)$ is the Kronecker's delta function. Assuming noiseless data, for N observations, K forms a $N \times N$ matrix given by:

$$K = \begin{bmatrix} k\left(x_1, x_1\right) & k\left(x_1, x_2\right) & ... & k\left(x_1, x_N\right) \\ k\left(x_2, x_1\right) & k\left(x_2, x_2\right) & ... & k\left(x_2, x_N\right) \\ . & & . & . \\ . & & . & . \\ k\left(x_N, x_1\right) & k\left(x_N, x_2\right) & ... & k\left(x_N, x_N\right) \end{bmatrix} \tag{1.22}$$

The objective of this exercise is to predict y^* at an input x^*. An assumption in GP modeling is that the joint data including x and x^* can be represented as a sample from a multivariate Gaussian distribution, therefore the joint posterior distribution yields:

$$\begin{bmatrix} y \\ y^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K & K^{*T} \\ K^* & K^{**} \end{bmatrix}\right), \tag{1.23}$$

where

$$K^* = \left[k\left(x^*, x_1\right) \quad k\left(x^*, x_2\right) \quad ... \quad k\left(x^*, x_N\right)\right], \tag{1.24}$$

and

$$K^{**} = k\left(x^*, x^*\right). \tag{1.25}$$

The conditional probability, $P\left(y^*|y\right)$ i.e. given y how likely is a certain prediction of y^*, follows a Gaussian distribution. Therefore by using Bayes's theorem the posterior distribution sampled from a Gaussian distribution is given by:

$$\left(y^*|y\right) \sim \mathcal{N}\left(\hat{\mu}, \hat{\sigma}\right), \tag{1.26}$$

where $\hat{\mu}$ is the mean of this distribution and is the best estimate is given by:

$$\hat{\mu} = K^* \cdot K^{-1} \cdot y, \tag{1.27}$$

and $\hat{\sigma}$ is the variance that captures uncertainty in the estimate given by:

$$\hat{\sigma} = K^{**} - \left[K^* \cdot K^{-1} \cdot K^{-1} \cdot K^{*T}\right], \tag{1.28}$$

The mean $\hat{\mu}$ of the distribution (1.26) can be thought of as the maximum-likelihood prediction for the output corresponding to the input x^*. In future sections the observations (x, y) are denoted as trial data because these data are used to supervise the model. It is important to note that an observation point x_i is a vector and therefore the independent variable, x, is a matrix for most of the realistic physical problems. The process and all the equations mentioned above are valid for the multi dimensional case. The prediction points (x^*, y^*) are denoted as test data because the GPR model is tested at these points.

Trial data D_{trial} is comprised of trial input data (X_{trial}) and trial output data (Y_{trial}). The objective is to predict test output data Y_{test} from test input data X_{test}. This is done by recovering the underlying process from noise observed data D_{trial} using the regression model described above. For a system with N_{trial} observations for trial data, a $K \times N_{\text{trial}}$ matrix is formed for X_{trial}:

$$X_{\text{trial}} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,K} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,K} \\ . & & . & . \\ . & & . & . \\ x_{N_{\text{trial}},1} & x_{N_{\text{trial}},2} & \cdots & x_{N_{\text{trial}},K} \end{bmatrix}, \tag{1.29}$$

where each row corresponds to a sample for K parameters, and each column corresponds to the N_{trial} observations for a specific input parameter. The computational simulation, which solves the equation, (1.14) is executed with X_{trial} to generate:

$$Y_{\text{trial}} = \begin{bmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,L} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,L} \\ . & & . & . \\ . & & . & . \\ y_{N_{\text{trial}},1} & y_{N_{\text{trial}},2} & \cdots & y_{N_{\text{trial}},L} \end{bmatrix}, \tag{1.30}$$

where Y_{trial} is the set of output parameters for the specific set of input parameters, X_{trial}, and L is the number of output variables. Similar matrices are defined for X_{test} and Y_{test} with the same dimensions K and L respectively.

1.4.3.3 1-D Illustration

Assume an experiment with 1-D data with $N = 6$ observations for an unknown function $y = f(x)$. The observations are at:

$$x = [-0.8, -0.5, -0.1, 0, 0.4, 0.75], \tag{1.31}$$

with a noise of $\sigma_n = 0.25$. Data (black stars) and the corresponding error (red bar) are presented in Figure.1.3. The exercise is to predict the value and error at $x^* = 1.0$ given by the blue dot in the figure. With a $\sigma_f = 1.27$ and $l = 1.0$, equation (1.21)

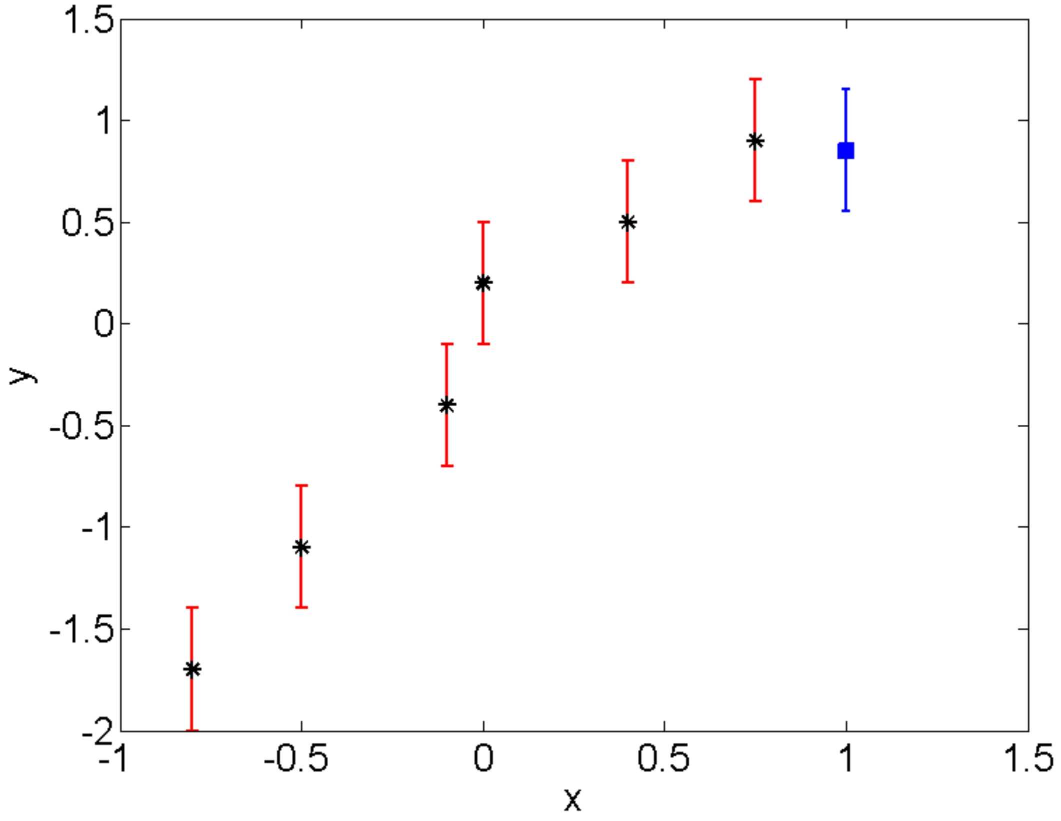


Figure 1.3: Sample 1-D data for regression

yields:

$$K = \begin{bmatrix} \mathbf{1.67} & 1.54 & 1.26 & 1.17 & 0.78 & 0.48 \\ 1.54 & \mathbf{1.67} & 1.49 & 1.42 & 1.07 & 0.73 \\ 1.26 & 1.49 & \mathbf{1.67} & 1.60 & 1.42 & 1.12 \\ 1.17 & 1.42 & 1.60 & \mathbf{1.67} & 1.49 & 1.21 \\ 0.78 & 1.07 & 1.42 & 1.49 & \mathbf{1.67} & 1.51 \\ 0.48 & 0.73 & 1.12 & 1.21 & 1.51 & \mathbf{1.67} \end{bmatrix} \tag{1.32}$$

Based on (1.25), the value for K^{**} computed using (1.31) is 1.67. Similarly based on (1.24), the value for K^{*} computed using (1.31) is $K^{*} = [0.32 \;\; 0.52 \;\; 0.88 \;\; 0.98 \;\; 1.34 \;\; 1.56]$. Using (1.27) and (1.28) the mean and error in the predictions are given by, $y^{*} = 0.885$ and $var\left(y^{*}\right) = 0.179$. Assuming noiseless data, if the above procedure is repeated for various points over the x-axis and multiple samples are drawn, the prior (1.19) looks like, where the model has randomly drawn three sample functions in the defined range. Similarly multiple samples drawn from the posterior 1.26 look like, In other words, the posterior (Figure 1.5) implies that the model has actually chosen the function those pass through the trial data set from a set of random infinite samples unlike the prior.

It is important to observe how the predicted values look for lots of data points. Instead of having error bars for each point, a 95% confidence interval $(y^{*} \pm 1.96\sqrt{var\left(y^{*}\right)})$ is drawn for the above problem and presented in Fig: 1.6.

1.4.4 Hyper-parameters and Kernel Function in Gaussian Processes

The kernel function in Eq.1.19 takes several forms, and so a judicious decision is taken to choose the form that fits the current sample data. The following standard forms are used in the research community:

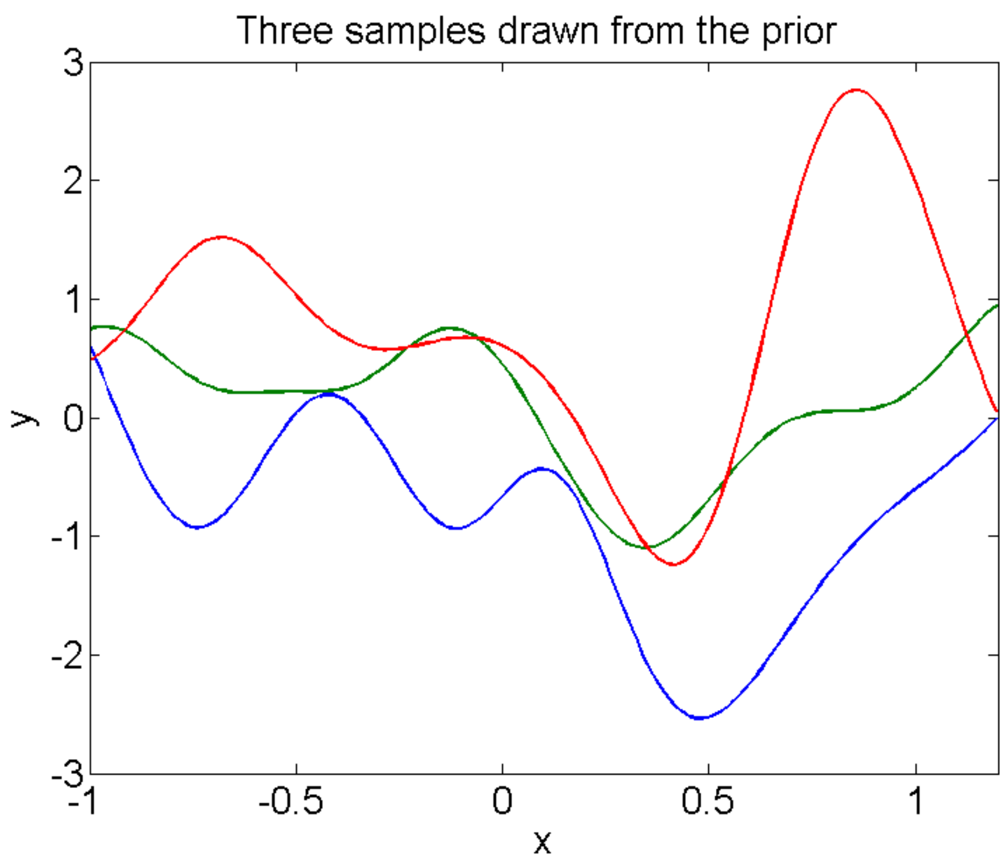


Figure 1.4: Three samples drawn for the prior

- Absolute exponential - Ornstein-Uhlenbeck Stochastic Model:

$$K(x,y) = \sigma_d \cdot \exp\left(-\sqrt{(x-y)' \cdot (x-y)}\right) \tag{1.33}$$

- Linear:

$$K(x,y) = \sigma_d \cdot x' \cdot y \tag{1.34}$$

- Squared exponential:

$$K(x,y) = \sigma_d^2 \cdot \exp\left(-\frac{(x-y)' \cdot (x-y)}{2 \cdot l^2}\right) \tag{1.35}$$

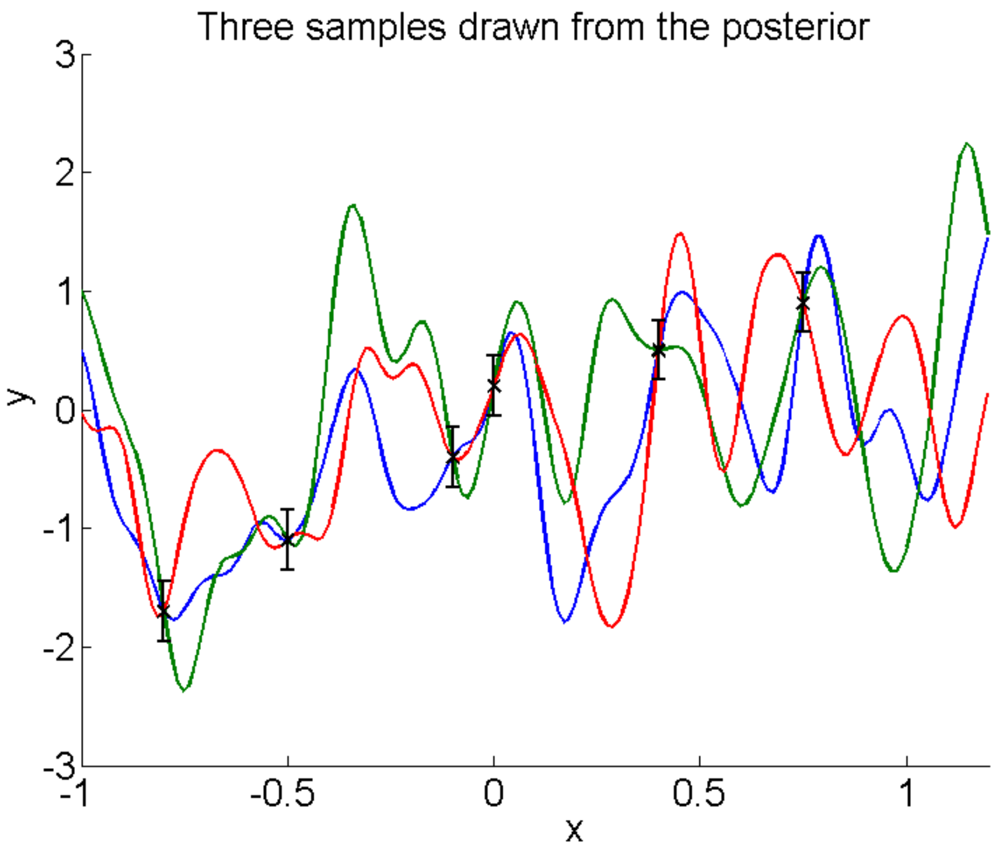


Figure 1.5: Three samples drawn for the posterior

- Periodic:

$$K\left(x,y\right)=\sigma_d^2\cdot\exp\left(-\sin\left(k\cdot\pi\cdot\left(x-y\right)'\cdot\left(x-y\right)\right)\right) \qquad (1.36)$$

The applications of the kernel function depend on the nature of training data. The variables σ_d, k, l are called hyper-parameters. A rigorous sensitivity study is performed to determine the best value for these hyper parameters. Before building the GPM, a choice for the kernel functions and the hyper-parameters is done based on cross-validation, and the final choice is used to fit the data and perform predictions.

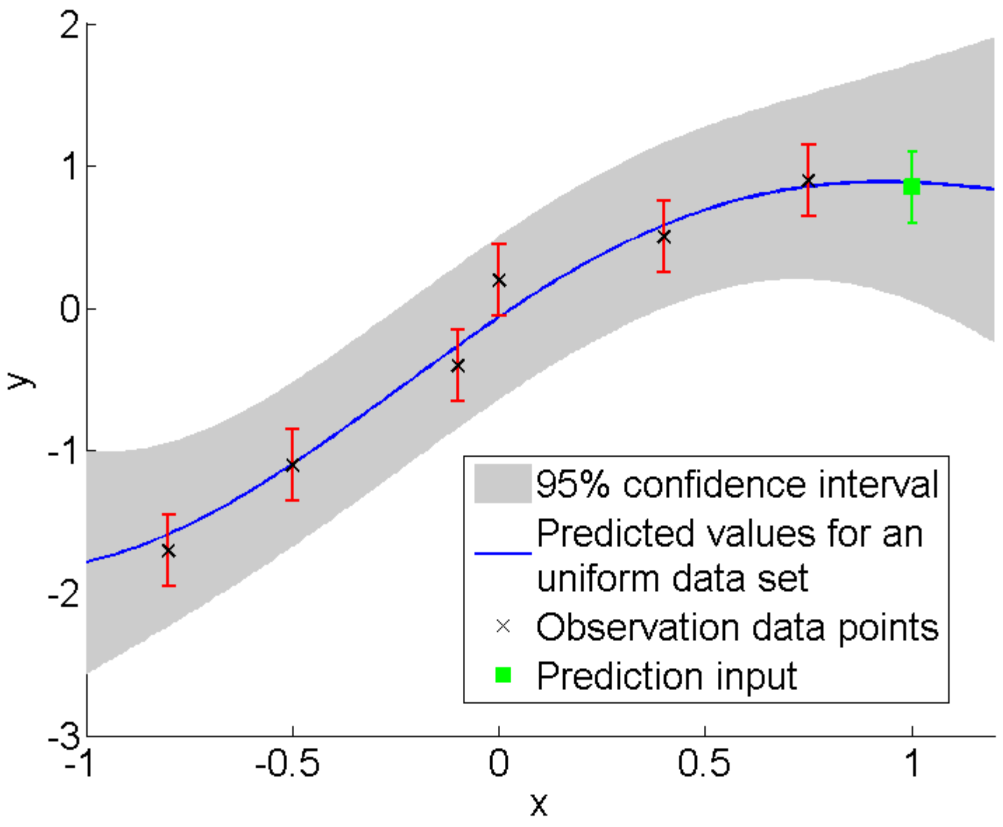


Figure 1.6: Prediction mean with confidence interval

1.5 Objectives and Tasks

The objective of this work is to develop and demonstrate an iterative optimization method using genetic algorithms and Gaussian process based regression for nuclear engineering applications using two reactor design problems as illustrative examples. The objective splits into two sub-objectives: Firstly to maximize or minimize the following group of functions:

$$F\left(\hat{X}\right) = f_1\left(\hat{X}\right) \wedge f_2\left(\hat{X}\right) \wedge f_3\left(\hat{X}\right) ...f_O\left(\hat{X}\right) \tag{1.37}$$

where $\hat{X}$ is a vector of input parameters, O is the number of optimization objectives, $f_o\left(\hat{X}\right)$ is an objective function based on the QOIs for an objective o. Operations are made for each of the individual functions. The function, $f_o\left(\hat{X}\right)$ can be a maximization or a minimization function. A standard method where the objective space is strictly convex is used for the weighted approach. The weighted approach presents the optimization problem in an intuitive and understandable form. In the weighted form the optimization problem is presented as a single function optimizer:

$$F\left(\hat{X}\right) = \sum_{o=1}^{O} f_o\left(\hat{X}\right) \cdot w_o \tag{1.38}$$

where the individual optimization functions, $f_o\left(\hat{X}\right)$ are converted to a consistent maximization or a minimization function. In the current formulation $\sum_{o=1}^{O} w_o = 1$. The nuclear reactor problem specification that defines the variables $\hat{X}$, and $f_o\left(\hat{X}\right)$ is presented in the next section. Secondly, characterization and definition of a nuclear reactor power system based on GCFBR design and a variant of the AP1000 design are presented, followed with a successful implementation of IHGOM on the GCFBR and AP1000 problems. Criticality, flux, and depletion calculations are performed using Serpent2 [31], HYBGASA using Java, and GPMEM using R. The detailed tasks in the design, development, analysis and validation of IHGOM include:

- Characterization of a GCFBR reactor power system and an AP1000 design as a complex system.
- Define a detailed set of optimization design parameters and objectives in the GCFBR reactor power system.
- Development of the iterative HYBGASA GOM method.

- Development of GPMEM regression system.
- Analysis of feasibility of IHGOM by testing if the iterative approach helps in reducing regression errors and is able to determine a near optimal solution.
- Compare the optimal solution obtained using the proposed method to a brute-force search. The comparison is made based on accuracy and speed, where speed is determined based on the number of actual executions of the complex system.
- Sensitivity studies are preformed to search for the best parameters for the iterative method.

1.6 Tools

Data analysis is done using Excel and R. The GPMEM models and validation methods are developed using R, Python and matlab. The optimization methods are developed and implemented in Java.

2. NOVEL OPTIMIZATION METHOD

This chapter presents the development details of the iterative method introduced in the previous section. This chapter includes the description of the MPE, fitness function, stopping criteria for GA, ϵ constraint method implementation, real GA operators implementation, choice of kernel and hyper-parameters in the emulator, and a detailed step by step approach of the problem execution.

2.1 Hybrid Optimization Method

Local search (A_{local}) is used to determine the optimal solution in the neighborhood of a solution. "1-Flip neighborhood" is a popular local search method used with GA. In the "1-Flip neighborhood" method,

- Each bit in the chromosome (s_i) is flipped to obtain a new solution ($\hat{s}_i$), and the fitness is calculated.
- If the fitness of $\hat{s}_i$ is better than the fitness of s_i, discard s_i, and $\hat{s}_i$ becomes s_i.

However, "1-Flip neighborhood" is not an effective local search method. Fitter solutions are sometimes ignored because, a solution could be non optimal locally but after operations like crossover and mutations, could emerge as a global optimal solution. Therefore instead of just discarding a chromosome with a lower fitness, a possible solution is to keep it alive with a non-zero probability. Therefore, the implementation of simulated annealing (SA) based concept along with "1-Flip neighborhood" as part of the local search comes into consideration. In SA, every neighboring solution is chosen with a positive probability. If $\hat{s}_i$ is a neighbor of s_i, the probability of accepting $\hat{s}_i$, $P(c_k)$ is given by,

$$P(c_k) = \begin{cases} 1 & : f(\hat{s}_i) >= f(s_i) \\ \exp\left(\frac{f(s_i)-f(\hat{s}_i)}{c_k}\right) & : f(\hat{s}_i) < f(s_i) \end{cases} \tag{2.1}$$

where $f()$ is the fitness, and c_k is the control parameter. If c_k is large, large changes in fitness are accepted with high probability. Similarly, when c_k is small, only smaller changes are accepted, and gradually as c_k approaches 0, no changes are accepted . This method of the implementation of a an hybrid GA and SA is called as the global optimization method (GOM).

2.2 Coupled GOM and GPM

GOM operates on the system as a black box with the goal of optimizing a set of design parameters to met a set of objectives. In the current implementation the system is a regression model instead of the actual physics based model. The regression model is built using a Gaussian processes based method (GPM). The input set from the chromosomes generated from the GA is fed into the GPM which, predicts the values for the QOIs. The GPM being a surrogate model has an inherent model error associated with every predicted QOI. This model error is called as the Model Prediction Error (MPE). The MPE has the inherent characteristics of being dependent on the size of the samples used to build the GPM. Therefore an iterative approach is implemented wherein the GPM is reconstructed by adding new samples iteratively to reduce the MPE. In this method the execution starts with the GPM built using samples from the whole design space, GA is executed on this GPM to obtain a pareto optimal solution, then new samples are generated at the vicinity of the optimal solution and the GPM is reconstructed with the new samples added to the old samples, then GA is executed on the new GPM and this process operates iteratively till an acceptable level of MPE is reached. Let $N_{initial}$ is the initial sample size used to train the GPM, N_Y is the number of outer iterations, and $N_{re-train}$ is the number of new samples generated after each outer iteration, the number of samples used every outer iteration to re-train the GPM is given by,

$$N_{outer,j} = N_{initial} + \left(N_{re-train} \cdot j\right), \tag{2.2}$$

where j is the outer iteration and $N_{outer,0} = N_{initial}$. A good value for $N_{re-train}$ is,

$$N_{re-train} = \frac{N_{initial}}{2}. \tag{2.3}$$

The multi-dimensional design space used in generating sampled using LHS is reduced by a re-sampling rate, $\Xi = 50\%$ after every outer iteration.

2.3 Multi-objective Optimization Development and Implementation

The global optimization problem presented in Eq.1.38 is given as,

$$\min_{X \in R^n} F(X) \tag{2.4a}$$

$$\text{constraints } G(X) \tag{2.4b}$$

$$x_{i,low} \leq x_i \leq x_{i,high} \tag{2.4c}$$

where $F\left(X\right)$ is a vector of objective functions, $G(X)$ is a vector of constraint functions, and X is a vector of input design parameters $x_{N_{var}}$. In a multi-objective domain $F\left(X\right)$ is not a single function, rather a group of functions. And assuming that the constraints are a group of functions, (2.4) in an expanded form is given by,

$$F\left(X\right)=\left\{f_1\left(x\right),f_2\left(x\right),.....,f_O\left(x\right)\right\} \tag{2.5a}$$

$$G\left(X\right)=\left\{g_1\left(x\right),g_2\left(x\right),.....,g_C\left(x\right)\right\}X=\left\{x_1,x_2,...,x_{N_{var}}\right\} \tag{2.5b}$$

where $f_o\left(x\right)$ is an objective function, $g_c(x)$ is a constraint function, O is the number of objectives, C is the number of constraints, $x_{i,low}$ is the lower limit for x_i, $x_{i,high}$ is the upper limit for x_i, and N_{var} is the number of design parameters. From (2.4) and (2.5), it is observed that minimization/maximization operator is applied on a group of objective functions, hence there is no possibility of obtaining a single optimal solutions, rather a group of equally optimal solutions called as the pareto optimality set. In a practical domain, qualitative judgment is used to determine the best among all the pareto optimal solutions. In the current work we desire to obtain atleast a single pareto optimal solution.

Fitness function (FF) is the measure of the proximity of the feasible solution to the pareto optimal solution. FF is the quantifiable scalar parameter that is used by GA to determine input sets in each generation. Determination of the fitness in a single objective function is straightforward i.e. if it is a maximization function, then the chromosome having a higher fitness value is more desired as compare to a chromosome with a lower fitness value. However, fitness function in a multi-objective system is complex and needs special treatment. The most generic solution is to convert a multi-objective function to a single objective function and use this as a fitness function. And, a very standard approach is the weighted approach where a weighted sum of all the objectives is constructed. Assuming a minimization objective function the optimization problem as a weighted sum is represented as,

$$\min_{X \in R^n} F\left(\hat{X}\right) = \sum_{o=1}^{O} f_o\left(\hat{X}\right) \cdot w_o \tag{2.6}$$

where $\hat{X}$ is a vector of input parameters, O is the number of optimization objectives, $f_o\left(\hat{X}\right)$ is an objective function based on the QOIs, and w_o is a positive weight for an objective o. In the current formulation s$\sum_{o=1}^{O} w_o = 1$. This method converts the multiple objective functions, $f_o\left(\hat{X}\right)$ to a single objective function, $F\left(\hat{X}\right)$. Every combination of this weighing factors generates a different pareto optimal solution. However this method has the following concerns,

- Determination of the weighting coefficients, w_o. In practical applications, qualitative judgement is used to determine the coefficients.
- The method assumes that the objective function space is convex. Therefore the problems where the shape of the objective space is not known the assumption of convexity is not accurate.

To avoid the complexities of the weighted approach in solving problems having non-convex objective space, the ϵ-constraint [32, 33] method is used.

2.4 Epsilon Constraint Method

The ϵ-constraint method is an effective method used in the optimization of multi-objective systems where the functional form of the relationship between the QOIs and the design parameters is not known. In the ϵ-constraint method, the multi-objective problem is reformulated into a single objective function, by using one of the objectives as the objective function, and the rest of the objectives as a grid based constraints. Different pareto-optimal solutions can be determined by changing the grid size. Assuming that there are three objectives given by, O_1, O_2, and O_3, the ϵ constraint method for the maximization/minimization of O_1 with the other two objectives, O_2 and O_3 used as constraints is presented in Fig: 2.1, The brown shaded

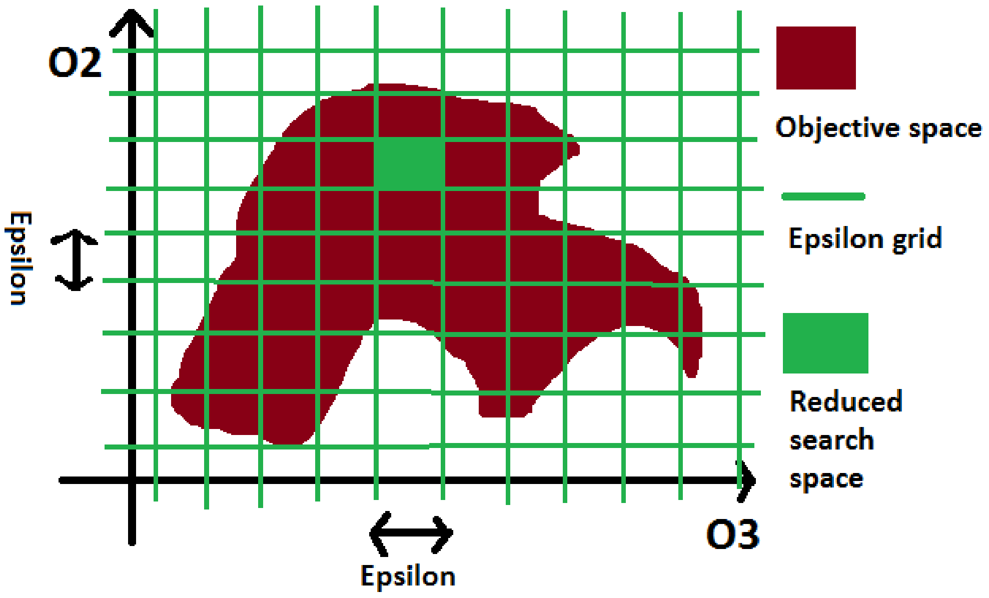


Figure 2.1: ϵ constraint method illustration. This is a hand sketch to illustrate the method.

region is the objective space nd the green grid lines represent the ϵ grids. Therefore the fitness function now depends only on O_1 in every grid. In simple terms, the global search is converted into several local searches to increase efficiency. In the ϵ constraint method the optimization problem, Eq.2.4 is reformulated as,

$$F(X) = f_o(x) \tag{2.7a}$$

$$G(X) = \{g_1(x), g_2(x), ..., g_C(x)\} \tag{2.7b}$$

$$\epsilon_{k,i,left} \leq f_i(x) \geq \epsilon_{k,i,right}, i = 1...O, i \neq o \tag{2.7c}$$

$$\sum_{0}^{K} k \in \Omega_o \tag{2.7d}$$

$$X = \{x_1, x_2, ..., x_{N_{var}}\} \tag{2.7e}$$

where k is the ϵ bin, K is the total number of ϵ bins. The minimization function 2.5a is now treated as,

$$\min_{X \in R^n} F(X) = \min_{X \in R^n} F_k(X), k = 1.....K \tag{2.8}$$

The objective is to maximize or minimize $(F(X))$ i.e. Eq.2.4. Using the ϵ constraint method, the multi-objective function (2.5a) is solved as a single objective function, (2.7a). (2.7b) are the actual constraints and (2.7c) are the new additional constraints. A simple illustration of the ϵ constraint based optimization method is presented as follows. Assume, there are three design variables: A,B and C. A multi objective demonstration problem is assumed as,

1. Maximize A
2. Minimize B
3. Maximize C

The epsilon method is demonstrated in the following steps,

1. Start with the *optimization objective of maximizing A*. Create several discrete spaces in B and C.
2. In each discrete space in B and C determine the maximum value for A. In the current case, the discrete space is 2-dimensional boxes.
3. Let A_{max} is the maximum value for A after searching all discrete B and C intervals.

4. Now set A=A_{max} and perform the following steps,
   (a) The *objective is to minimize B*. Create discrete intervals for C. This forms a vector with intervals of C.
   (b) For each interval of C, and with A=A_{max}, determine the minimum value for B.
   (c) Let B_{min} is the minimum value for B after several searches for all the intervals of C with A=A_{max}
      i. Now with A=A_{max}, B=B_{min}, *search for the maximum value for C* i.e. C_{max}

The optimum values are A_{max}, B_{min} and C_{max}. The italicized text in the above steps are single objective optimization problems. It is observed that in the ϵ-constraint method, a multi objective problem for A,B and C is converted into an iterative single-objective problem i.e. each of the objectives in the multi-objective case is treated separately as multiple single-objective cases. Different Pareto optimal solutions are obtained by changing the sequence of the objectives in the iterations.

2.5 GA Operators

In practical physics based problems when the design parameters are continuous the real valued GA is implemented. The implementation of real valued GA is similar to the binary GA with the primary difference being the fact that variables are no longer represented by bits of zeros and ones, but instead by floating point numbers. Real valued GA requires less storage as compared to binary GA because to obtain a desired precision in the real numbers using bits, the chromosome size needs to be significantly large. Due to not having decoding and encoding steps, the real valued GA is faster than binary GA. Real valued GA follows the same heuristics operations however, there is a significant difference in the implementation of the operators. The operators are implemented in the following way,

2.5.1 Variables

In real valued GA, the chromosome is defined as an array of variable values of the design parameters. If there are N_{var} design parameters, the chromosome is used as an array of size, N_{var}.

$$chromosome = [p_1, p_2, p_3, \ldots.., p_{N_{var}}] \tag{2.9}$$

where p_i is the design parameter of index i. The fitness function can be defined as,

$$F = f\,(chromosome) = f\,(p_1, p_2, p_3, \ldots.., p_{N_{var}}) \tag{2.10}$$

Fig: 2.2, presents an approach sequence flowchart of real valued GA.

2.5.2 Initial Population

The initial population is generated using the LHS method for all the design parameters and the chromosomes in the variable value range. The random number generator, generates a random floating point number between 0 and 1.0. The input design parameter is therefore scaled to its range using the following equation,

$$p = p_{low} + rand() \cdot (p_{high} - p_{low}) \tag{2.11}$$

The initial population gives rise to $N_{pop} X N_{var}$ matrix with random numbers in them. LHS sampling plays a very important role in generating this widely distributed samples.

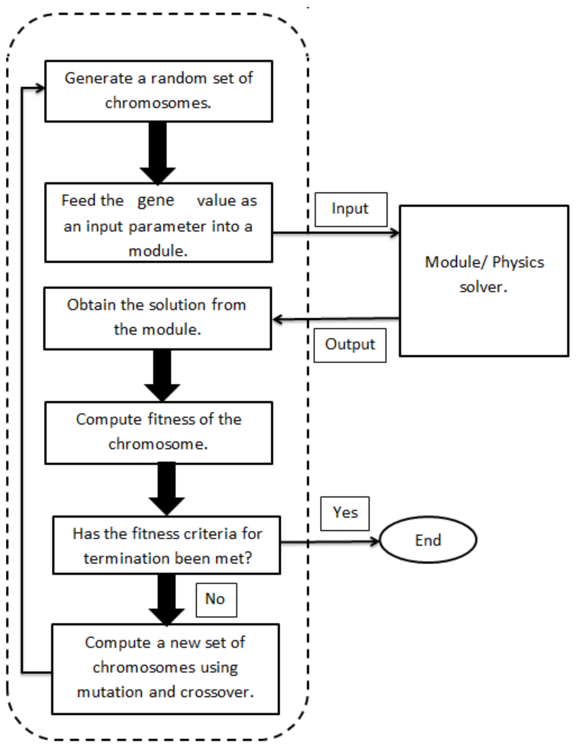

Generate a random set of chromosomes.
Feed the gene value as an input parameter into a module.
Input
Module/ Physics solver.
Obtain the solution from the module.
Output
Compute fitness of the chromosome.
Has the fitness criteria for termination been met?
Yes
End
No
Compute a new set of chromosomes using mutation and crossover.

Figure 2.2: A macroscopic view of the import components in genetic algorithms

2.5.3 Selection

The selection methods in real valued GA are similar to those used in binary GA. The roulette wheel selection method is employed to select the parent chromosomes for crossover and mutation. First, each of the chromosome is associated with a probability based on its rank and then a roulette wheel is built based on the probabilities. Then a random number is generated and the corresponding chromosome from the roulette wheel is selected.

2.5.4 Crossover

Crossover on the parents selected in a binary GA is straight forward, but it is not so intuitive when it comes to real valued GA. It is desired to have a crossover operator such that it searches for off-springs based on the distance between the parent solutions. A popular method, “simulated binary crossover” (SBX)[32] is widely accepted in the optimization community. In the SBX method, two parent solutions create two off-springs. Let, $x_i^{1,p}$, $x_i^{2,p}$ are the parent chromosomes of design parameter, i, the SBX method generates the off-springs $x_i^{1,o}$, $x_i^{2,o}$. A spread factor, β_i defined as,

$$\beta_i = \left| \frac{x_i^{2,o} - x_i^{1,o}}{x_i^{2,p} - x_i^{1,p}} \right| \tag{2.12}$$

is the ratio between the absolute values of the difference between the parents and the difference between the off-springs. The factor, β_i is obtained from a specified probability distribution function such that the area under the probability curve from 0 to β_i is equal to a random number u_i. The probability distribution is given as,

$$P\left(\beta_i\right) = \begin{cases} 0.5\left(\eta+1\right)\beta_i, & \text{if } \beta_i \leq 1;. \\ 0.5\left(\eta+1\right)\frac{1}{\beta_i^{\eta+2}}, & \text{otherwise.} \end{cases} \tag{2.13}$$

where η is a tuning parameter. A large value for η gives a higher probability for creating near-parent solutions and a small value for η allows distant solutions to be created as off-springs. Equating the area under the probability curve to u_i, the ordinate of the function $\beta_{q,i}$ yields,

$$\beta_{q,i} = \begin{cases} \left(2u_i\right)^{\frac{1}{\eta+1}}, & \text{if } u_i \leq 0.5;. \\ \left(\frac{1}{2(1-u_i)}\right)^{\frac{1}{\eta+1}}, & \text{otherwise.} \end{cases} \tag{2.14}$$

Based on the values of $\beta_{q,i}$, the off-springs are generated as,

$$x_i^{1,o} = 0.5\left[\left(1+\beta_{q,i}\right)x_i^{1,p} + \left(1-\beta_{q,i}\right)x_i^{2,p}\right] \tag{2.15}$$

$$x_i^{2,o} = 0.5 \left[(1 - \beta_{q,i})\, x_i^{1,p} + (1 + \beta_{q,i})\, x_i^{2,p} \right]. \tag{2.16}$$

Practically, a value for η is chosen, a random number for u_i is generated, $\beta_{q,i}$ is calculated using 2.14, and then the off-springs are calculated using 2.15. Fig: 2.3 presents the distribution for the offsprings based on two values for η. The plot shows the distribution based on two values for η (Figure obtained from ([32])) Fig: 2.4

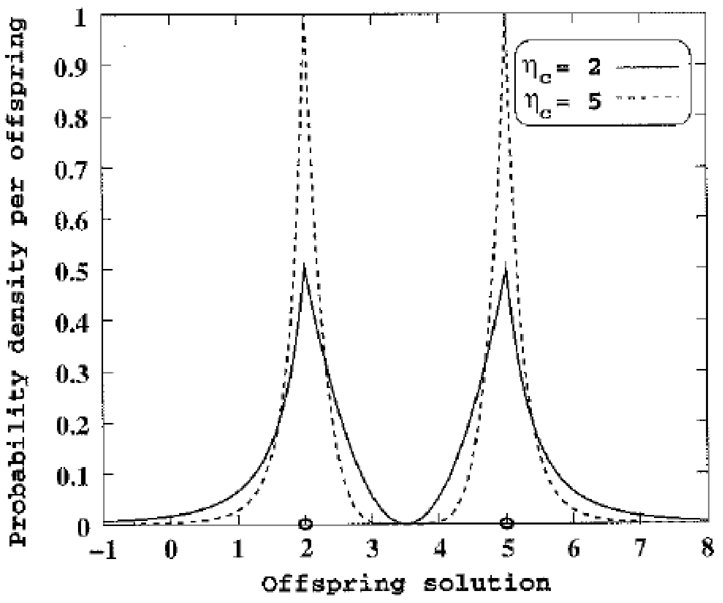


Figure 2.3: The probability distribution function for creating offspring using the SBX method. Parents are at x-location "2" and "5"

presents the distribution with respect to the distance between the parents. The SBX method is useful because the difference between the off-springs is proportion to the parent solutions, and near parent solutions are more likely to be chosen as off-springs as compared to distant solutions.

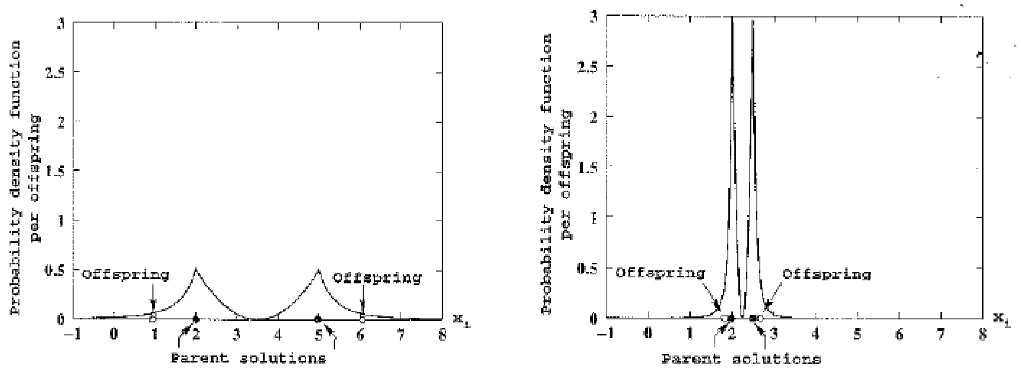


Figure 2.4: The probability distribution function with two different set of parents for $\eta = 2$. Left: the parents are at 2 and 5, Right: the parents are closer at 2.0 and 2.5. (Figure obtained from [32])

2.5.5 Mutation

In real valued-GA the mutation operator is straightforward and generates a random number in the value range for a specific parameter in a specific chromosome in the population.Mutation is operated on the non-elite members of the population. The total number of mutations to be performed after cross-over operation is given by,

$$N_{mut} = N_{var} * N_{chromo} * \nu_{mut} \tag{2.17}$$

where N_{mut} is the number of mutations in the population, N_{var} is the number of design parameters, N_{chromo} is the number of chromosomes in the population, and ν_{mut} is the mutation rate. A higher mutation rate ensures a higher diversity in the search, however it makes the convergence slow. In contrast, a lower mutation rate reduces the diversity in search.

2.6 Stopping Criteria for GA

The optimum solution of the problem is not known and therefore the search can go on for an infinite amount of time which is practically not feasible. Therefore, a standard approach is to analyze the solution after every generation and develop a stopping criteria based on the fittest value in that generation, the variance in data in the population and the change in the fittest value with generation. All the three conditions have to be met to conclude GA iterations and accept the resulting solution as the desired pareto optimal solution. The three parameters are,

2.6.1 impFittest (τ_{IF})

This variable keeps track of the improvement in the best fitness in a generation. At the end of each generation the change in the fittest fitness value i.e.τ_{IF} is calculated. During the initial generations, τ_{IF} carries a larger number which decrease with generations, and as the iterations are closer to achieving the optimal solution, τ_{IF} reduces considerably. The iterations are continued till τ_{IF} reaches an user defined threshold. If this is the only stopping criteria for a minimization problem, then the iterations are stopped when,

$$\tau_{IF} < th_{IF}. \tag{2.18}$$

2.6.2 maxRelDistance (τ_{MRD})

This variable keeps track of the maximum distance between each chromosome and the fittest chromosome. If f_i is the fitness of chromosome i and f_{best} is the fitness of the best chromosome,

$$\tau_{MRD} = \frac{\max\left(|f_i - f_{best}|\right)}{f_i}, i = 1...N_{chromo} \tag{2.19}$$

2.6.3 relDiff ($\tau_{relDiff}$)

Relative difference between f_i and f_{best} is given by,

$$\tau_{relDiff} = \frac{|f_{mean} - f_{best}|}{f_{best}}, \tag{2.20}$$

and the stopping criteria is that, a generation should have a $\tau_{relDiff} < d$, where $d_{relDiff}$ are tuning parameters.

2.7 Model Prediction Error Estimation

There are several ways to quantify MPE. Since the actual value for the independent variable for the test data is not available a direct residual or a difference is not possible to compute and if possible would not make sense. Therefore there are standard methods used to actually quantify MPE based on the available observations and the predictions. Following MPE methods are used in this work to understand and quantify the quality of the predictions,

- K-Fold cross validation: In the K-fold cross validation method, the inputted data set of size N is split into K folds of $\frac{N}{K}$ samples in each fold. In other words, K experiments are performed with each experiment has N_c samples comprising of the $K-1$ folds is used as the training set and N_v samples comprising of one fold is used as the test set, and $N = N_v + N_c$. The N_c data points are used to fit the model and N_v data points are used to quantify the predictive ability of the model. Predictions are done for all the N_v data points and then these are compared with the actual N_v observations and a root mean square (RMS) of the model error is determined. This RMS of error is used as the MPE in the current problem.

  The determination of the value for K is an important trade-off between accuracy and speed. With a large number of folds, the bias of the error will be small, however the variance of the error would be large and the computation time is significantly large. However, with a small number of folds the number of validation experiments are reduced, hence the computation time is reduces, but the error in estimation is large. Therefore, there is a trade-off between speed and accuracy. Apparently, in very large data sets, upto 10-Fold CV is acceptable, but for sparse datasets and to be conservative the N-Fold cross-validation is employed. The N-Fold cross-validation is called as the Leave one out cross validation (LOOCV).

- Leave One Out Cross Validation (LOOCV): LOOCV is a special case of the K-Fold cross validation method. In the LOOCV case, $K = N$ or in other words, $N_v = 1$. In the LOOCV method each sample from the data set is "left out", the regression model is trained using the rest of the samples, and then this "left out" sample is tested against the model. This is the most conservative approach, but is the most accurate error predictor. A root mean square of all the errors are calculated to be used as the final MPE for the model with the specified training data set.

2.8 Selection of Kernel Functions and Hyper-parameters

The choice of the kernel function and the optimum value for the hyper parameters is important for the accuracy in the predictions from the GPM. A grid based cross-validation (GridCV) method is employed to make the choice for the kernel functions and the hyper-parameters. In the GridCV method, a grid based on several different options and values for the hyper parameters is developed and for each entry in the grid, the surrogate model is built and cross-validation is performed on the training set. Cross-validation results in the MPE based on the RMS values of the error due to each prediction in the cross-validation steps. Based on the MPE for values for each of the elements in the grid the final kernel function and the values for the hyper-parameters. Following grid is formed that determines the MPE for several combinations among the equations presented in Section:1.4.4. All the four options with several values for the hyper parameters, σ_d have been used. The search is performed and the best kernel function and hyper-parameter obtained is used for a specific QOI in a particular outer iteration. It is to be noted that the search performed after every outer iteration because the training set used to fir th model is updated at the beginning of every outer iteration. Each QOI has its own relationship with the design parameters, therefore the search is performed separately for each QOI.

2.9 Regression with Adaptive Feature Set

Regression with Adaptive Feature Set (RAFS) is the applicable to problems where the predictors are dependent on time. This is different from forecasting due to the fact that in RAFS the feature set is updated while in forecasting, the same feature set is used for predictions but with updated temporal data. RAFS is applicable in reactor design problems where the QOIs are dependent on the burn steps. In these type of problems there are the “base ” predictors and “burn” dependent predictors. The prediction model for QOI’s in the first burn step use the “base ” predictors only. In the prediction model for the QOI in the second burn step the predictor set consists of the “base” predictors as well as the QOI of the first burn step. Similarly, in the prediction model for the QOI in the third burn step the predictor set consists of the “base” predictors as well as the QOI of the first and the second burn steps. Therefor the predictor set is updated after every burn step. This method is implemented in GPM predictive model for the MAP1000 demonstration problem. The RAFS method is explained as, A predictive model with P predictor variables and D dependent variables is defined as,

$$y_d = F\left(x_P\right) \tag{2.21a}$$

$$0 < d \leq D \tag{2.21b}$$

$$0 < p \leq P \tag{2.21c}$$

where $d$and p are indices for the individual predictor and dependent variables. Using the RAFS method the predictive model for the first dependent variable y_1 is,

$$y_1 = F\left(x_1, x_2, x_3....x_P\right). \tag{2.22}$$

The second dependent variable y_2 is,

$$y_2 = F\left(x_1, x_2, x_3....x_P, y_1\right). \tag{2.23}$$

Similarly, the third dependent variable y_3 is,

$$y_3 = F\left(x_1, x_2, x_3....x_P, y_1, y_2\right). \tag{2.24}$$

And, the predictive model for the last dependent variable y_D is,

$$y_D = F\left(x_1, x_2, x_3....x_P, y_1, y_2....y_{D-1}\right). \tag{2.25}$$

This shows that the training model for y_D not only gets contributions from x_i but also gets contribution from other preceding dependent variables, $y_i...y_{D-1}$. This feature is the novelty in the way the predictive model is built.

2.10 Solution Flow

A schematic sketch of the research flow is presented in Fig: 2.5. The inner loop uses HYBGASA to perform a global optimization of the input parameters, and determines a neighborhood of the optimum space based on a defined set of objectives. The outer loop performs a reduction of regression errors in the predictions by updating the trial input set used to build the GPMEM. The initial trial input

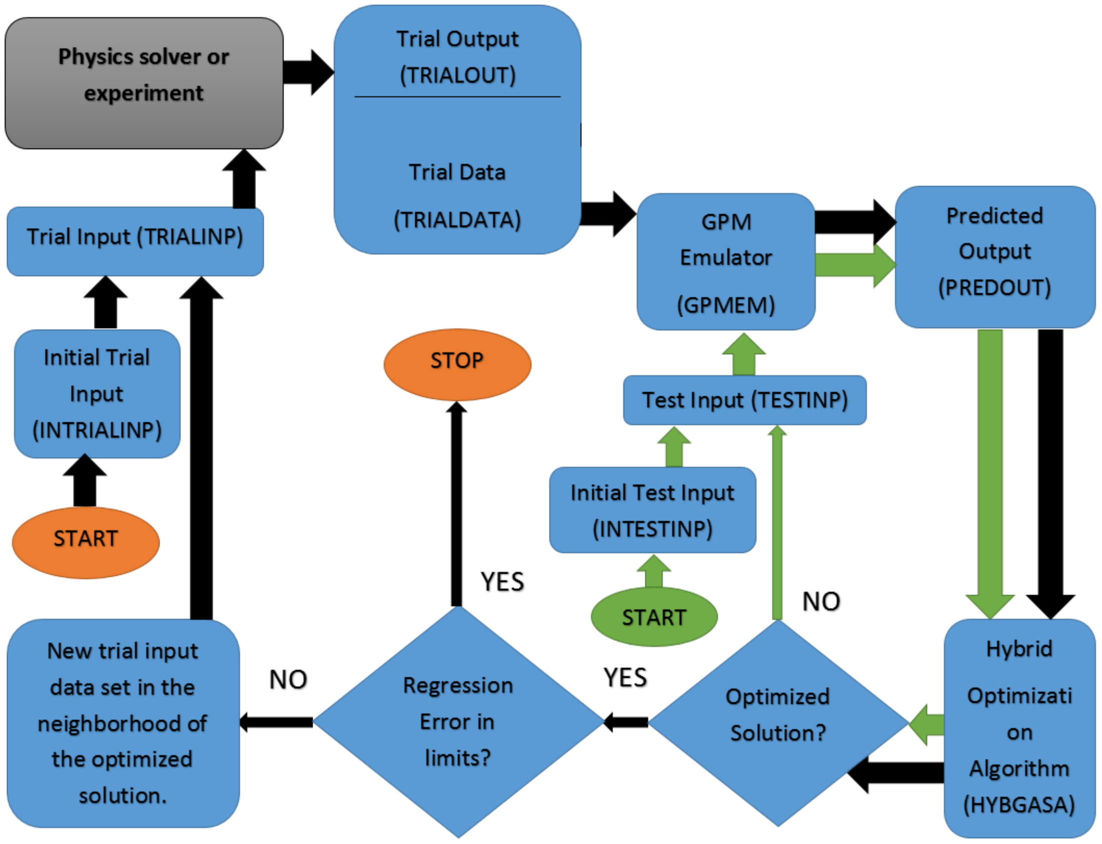

Physics solver or experiment
Trial Output (TRIALOUT)
Trial Data (TRIALDATA)
Trial Input (TRIALINP)
GPM Emulator (GPMEM)
Predicted Output (PREDOUT)
Initial Trial Input (INTRIALINP)
STOP
Test Input (TESTINP)
Initial Test Input (INTESTINP)
START
YES
START
NO
New trial input data set in the neighborhood of the optimized solution.
NO
Regression Error in limits?
YES
Optimized Solution?
Hybrid Optimizati on Algorithm (HYBGASA)

Figure 2.5: Research flow (Black: Outer loop, Green: Inner loop)

set (INTRIALINP) is obtained using the Latin hypercube sampling (LHS) method [34], [35]. During each iteration of the outer loop the trial set is updated with new data from the neighborhood of the near optimal solution. The emulator is re-trained using the updated trial-set. The inner loop performs HYBGASA using predictions from the GPMEM to determine the near optimal neighborhood. The solution flow can be summarized in the following steps,

1. Obtain the initial trial data by executing the complex system with a set of random samples developed using LHS.
2. Perform sensitivity studies to determine an ideal value for ϵ.
3. perform cross-validation and determine an optimized set of hyper-parameters in the Gaussian process based regression.
4. Build the emulator.
5. Start the epsilon loop and determine ϵ constraint bins.
6. Start with a bin from the ϵ grid,
7. Start GA and perform optimization to determine an optimal solution in the ϵ grid.
8. Perform Step-8 for all the elements in the ϵ grid. Once all the grids are exhausted,
9. Re-sample at the vicinity of the optimal solution and go back to Step-2.
10. Converge the outer iteration

3. METHOD DEMONSTRATION

This chapter presents a detailed description of results and analysis of the optimization problems solved using the methods developed in this dissertation. The section begins with a description and demonstration of the optimization method's capability on the Ackley's test problem. Next, descriptions of the sample systems are presented, including the parameters and the objectives of the optimization problem. Following the description is, a detailed presentation and an analysis of the results. At the end of each demonstration, a discussion on the novelty of the method is included, based on speed, accuracy and optimality. In a macroscopic view, the following analysis is made for the sample problems:

- Speed: The speed is determined by a comparison of the execution times of the converged iterative method and a brute force search when determining the optimum values for the design parameters. A comparison between local re-sampling and global re-sampling is done to analyze the speed in the reduction of MPE due to the outer iterations.

- Accuracy: Accuracy of the iterative method is determined by comparing the solutions with the solution obtained from the actual experiment or the system if it is a simulation.

- Search efficiency: Search efficiency is analyzed by comparing the optimum solution obtained using the iterative method and the best solution obtained using brute force search. The proximity of the obtained solution to the objectives and the constraints are analyzed. The Ackley benchmark test is employed to determine the search capability of the HYBGASA method.

- Sensitivity study: Sensitivity study analysis used to determine the values for the hyper parameters in GA and GPM.

The problems are designed to demonstrate the proposed optimization method. The problems are multi-objective problems with a blend of individual maximization, minimization and limit single objective functions. Some of the design parameters, QOIs and objective functions might not be applicable in the physical world, however in the current demonstration those are necessary to validate the proposed method.

3.1 Ackley Test Problem

The Ackley function [36, 37, 38] is widely used in testing and benchmarking optimization algorithms. It is a single optimization test function and is characterized by having several local minima and a single global minima.

3.1.1 Problem Description

The function is given by,

$$f(X) = -a \cdot \exp\left(-b \cdot \sqrt{\frac{1}{d} \cdot \sum_{p=1}^{P} x_p^2}\right) - \exp\left(\frac{1}{d} \cdot \sum_{p=1}^{P} cos(c \cdot x_p)\right) + a + \exp(1) \quad (3.1a)$$

$$-5.0 \leq x_p \leq 5.0 \quad (3.1b)$$

$$\text{minimum at } f(0, \cdots, 0) = 0 \quad (3.1c)$$

where P is the number of design parameters, a,b and c are constants, X is the input values. For $P = 2$, X is a 2-dimensional matrix. The global minima is at $X = 0$ i.e. $f(0) = 0$. The objective is to determine the global minima using the developed optimization method. Fig: 3.1 is the solution for the Ackley function 3.1 with the following values for the parameters, Table: 3.2 presents the parameters used in the GA optimization search.

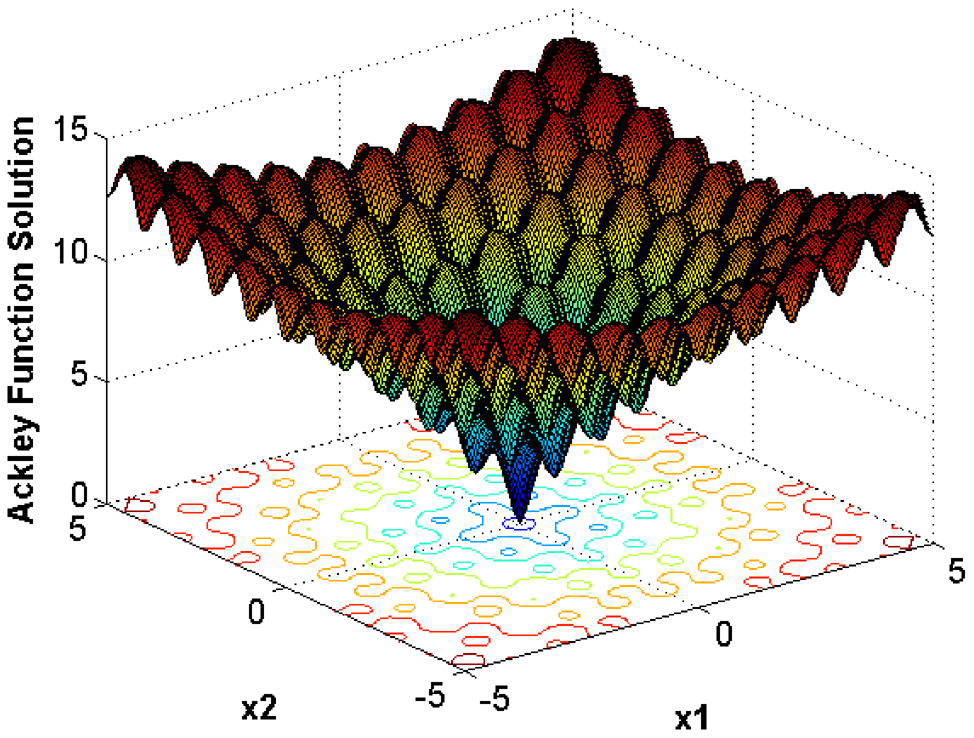


Figure 3.1: Ackley test problem solution

Parameter	**Value**
P	2
a	20.0
b	0.2
c	2.0π

Table 3.1: Ackley test parameters

3.1.2 Results

GA is used to determine the global minima. Fig: 3.2 is the solution for the Ackley function 3.1 converging towards a global minima. The image at the top presents the best chromosome in each generation and how the input values converge towards the

	Parameter	**Values**
GA	Number of chromosomes	200
	η	0.05
	ν_{mut}	0.5
	c_k	0.25

Table 3.2: Ackley optimization parameters

optimal input. The image at the bottom of the figure shows the solution of Ackley function for the best chromosome in each generation. The figure shows the convergence for the Ackley's function towards the global minima. The numbers shows the convergence towards machine precision. Fig: 3.3 is the distribution of chromosomes in the first and the last generation. In the first generation the search is random and the chromosomes are randomly distributed in the space. As the search moves towards convergence the randomness in the chromosomes is gone. This is evident from the bottom figure that represents the chromosome distribution in the last generation. The search capability of HYBGASA in single objective optimization problems is demonstrated. In the developed ϵ-constraint method all the single-objective problems are solved using HYBGASA. The objective of this exercise is to determine the global minima in the presence of several local minima. The method has been very effective in searching for the global minima at a significant speed.

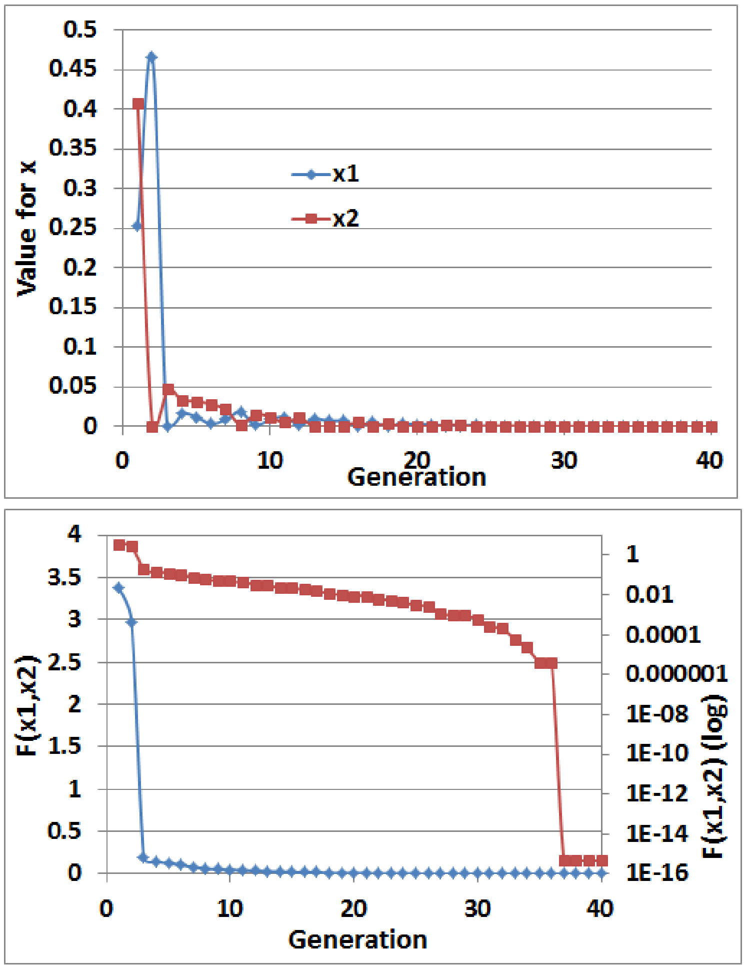


Figure 3.2: Ackley optimization search versus generation. Top: Input for the best chromosome, Bottom: Solution for the Ackley function for the best chromosome

3.2 Asymmetric 2-D Test Problem

An asymmetric 2-D problem is used to test the search efficiency of the optimization method. Along with the Ackley's problem, the asymmetric 2-D problem form an effective test and benchmarking tool for search algorithms. The objective is to search for the global maxima and minima on a surface.

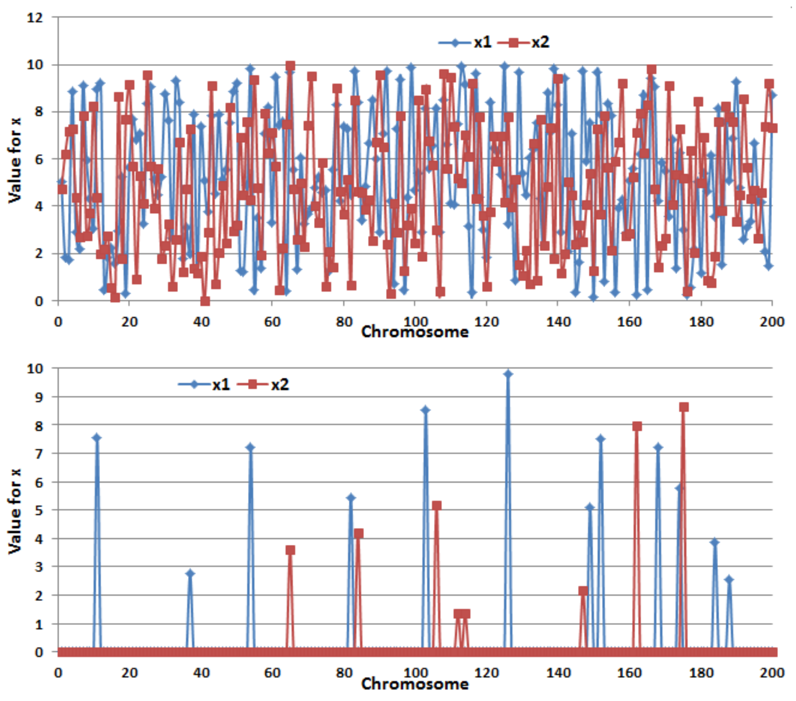


Figure 3.3: Top: Chromosome distribution in the first generation, Bottom: Chromosome distribution in the last generation.

3.2.1 Problem Description

The function is an additive function comprising of exponential and sine functions. The function is given by:

$$\begin{aligned} f\left(x\right) \quad = \quad & 1.3356[1.5\left(1-\hat{x}_1\right)+ \\ & \exp\left(2\hat{x}_1-1\right)\sin\left(\left(3\pi\left(\hat{x}_1-0.6\right)^2\right)\right)+ \\ & \exp\left(3\hat{x}_2-1.5\right)\sin\left(\left(4\pi\left(\hat{x}_1-0.9\right)^2\right)\right)], \end{aligned} \tag{3.2}$$

$$x_{1,2} \in \left(-5.0, 5.0\right), \tag{3.3a}$$

$$\hat{x}_{1,2} = \frac{x_{1,2}}{10} + 0.5, \tag{3.3b}$$

where $x_{1,2}$ are the input variable and $f(x)$ is the surface. Fig: 3.4 presents a discrete representation of the surface using 1000 discrete meshes in the input parameters.

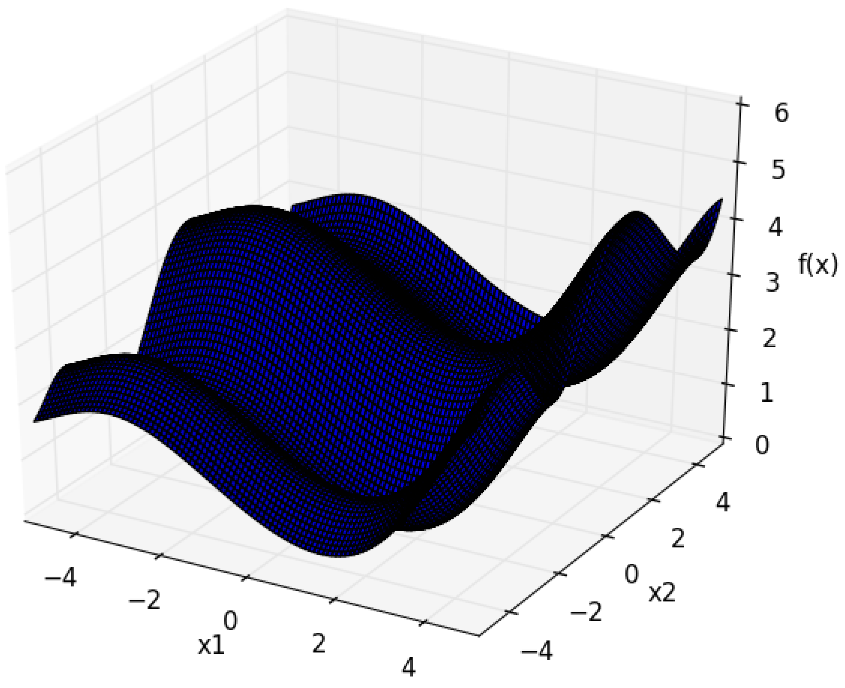


Figure 3.4: Asymmetric 2D problem surface (1000 mesh size).

3.2.2 Results-Maxima

The global maxima is determined using the optimizer, and a discrete search is done using a mesh of size 4000. The optimizer and discrete solutions are compared to an analytical solution by computing the relative error. Table: 3.4 presents the parameters used in the GA optimization search. THE GA solution and convergence is presented in Fig: 3.5

Solution Type	**x1**	**x2**	**f(x)**	**f(x) rel error**
Our method	5.0	$9.0657E-01$	5.2589	$5.5145E-11$
Discrete	5.0	$9.0648E-01$	5.2589	$4.6588E-10$
Analytic	5.0	$9.0655E-01$	5.2589	

Table 3.3: Solution for the maxima of a-symmetric 2d test problem

	Parameter	**Values**
GA	Number of chromosomes	200
	Number of generations	10
	η	0.05
	ν_{mut}	0.5
	c_k	0.25

Table 3.4: Asymmetric 2d test problem optimization parameters

3.2.2.1 Discussion

The GA search with 10 generation and 200 chromosomes in each generation, amounting to 2000 execution of the function has performed better that the brute-force search of 4000 discrete operations. The tuning parameters applied to the Ackley's problem has been used in this exercise.

3.2.3 Results-Minima

The global minima is determined using the optimizer. The solution is compared to a discrete solution obtained using a mesh size of 4000. Table: 3.6 presents the

Solution Type	**x1**	**x2**	**f(x)**
Our method	1.55338	-1.99490	$2.30473E-02$
Discrete	1.55414	-1.99424	$2.30476E-02$

Table 3.5: Solution for the global minima of a-symmetric 2d test problem

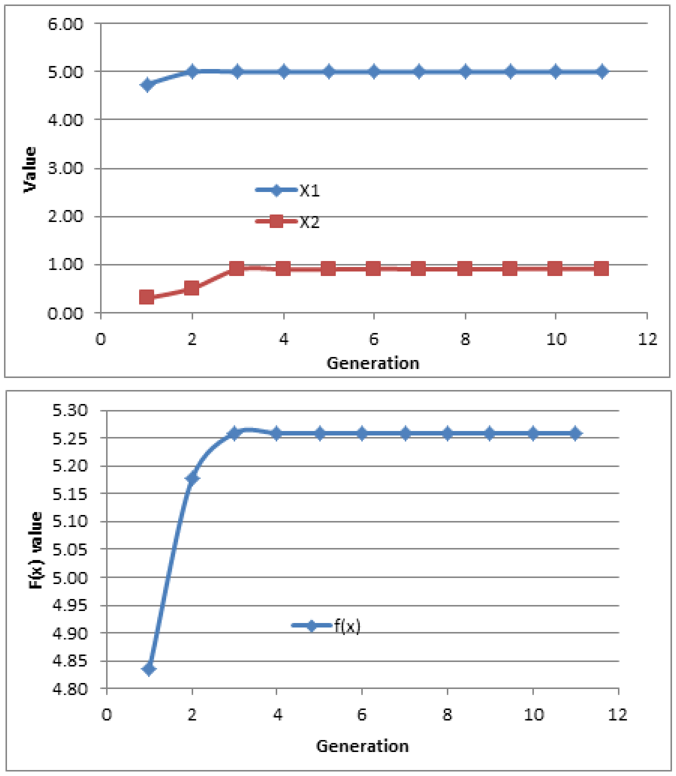


Figure 3.5: Top: convergence of x, bottom: convergence of f(x)

parameters used in the GA optimization search. THE GA solution and convergence

	Parameter	**Values**
GA	Number of chromosomes	200
	Number of generations	10
	η	0.05
	ν_{mut}	0.5
	c_k	0.25

Table 3.6: Asymmetric 2d test problem (minima) optimization parameters

is presented in Fig: 3.6

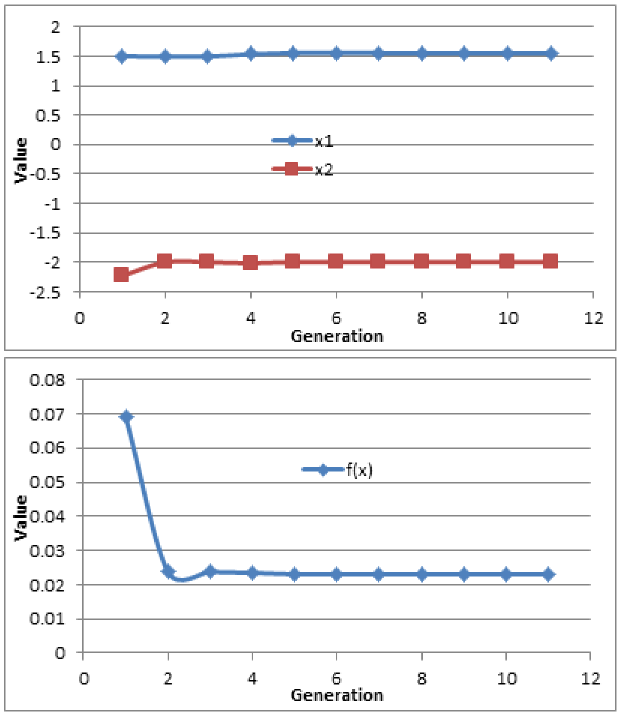


Figure 3.6: Top: convergence of x, bottom: convergence of f(x)

3.2.3.1 Discussion

The GA search with 10 generation and 200 chromosomes in each generation, amounting to 2000 execution of the function has performed better that the brute-force search of 4000 discrete operations. The tuning parameters applied to the Ackley's problem has been used in this exercise.

The objective of this exercise is to determine the global maxima and miniima in the presence of several local peaks. The method has been very effective in searching for the global maxima and minima at a significant speed.

3.3 Gas Cooled Fast Breeder Reactor Optimization

In this section, a detailed description of the reactor design including the neutronics and thermal-hydraulics aspects is presented. The Gas Cooled Fast Breeder Reactor (GCFBR) is a Helium cooled, fast breeder reactor. The fuel for this reactor is comprised of thorium, and the blanket is made of used LWR fuel. The objective of this problem is to determine an optimum set of values that would allow the core design to obtain criticality and minimize the peaking factors, with an acceptable pressure drop and coolant temperature in the core. The constraints are defined such that the reactor remains critical, and the pressure drop and the peaking factors are within limits. The first step described below is to develop a reactor model with thermo-fluid analysis and an energy transfer module to simulate a complex system. Then, the proposed method is implemented for the optimization problem for the system. This section concludes with a description and analysis of the results.

3.3.1 System Definition

The following physics based modules have been determined to perform the analysis required for the research. The first module, employed is the fuel pin cell module that determines the infinite neutron multiplication factor, KINF (KINF) based on the radius (RADIUS), as well as the isotopic enrichment (ENRICH) of the fuel element. Therefore, the input parameters defined in this module are RADIUS, and ENRICH, and the output parameter is the KINF. Included in this module, is the design of the whole core of the reactor. The output parameters in this module are the radial peaking factor (RADPF), and neutron multiplication factor, KEFF. This module has the same input parameters as defined in the first module. The second module, is a basic thermal-hydraulics and heat transfer module, where a hot channel analysis is performed to analyze the heat transfer across the fuel pin cell i.e. the flow of heat from the fuel pin to the coolant. The input parameters includes the: coolant temperature at the core inlet (TIN), the coolant mass flow rate (W) to determine the core outlet temperature, T_{out} (TOUT), the maximum fuel temperature (TMAX), and the pressure drop (DELTAP) across the flow channel. In this module the input parameters are, TIN, W, and ENRICH, and the output parameters are TOUT and DELTAP. The third module is the energy transfer module; it performs a basic Brayton's cycle calculation to determine EFF. The input parameters defined in this module are TOUT and TIN, and the out parameter is EFF. A detailed description of all of the parameters are presented in [39].

3.3.1.1 Reactor Design

This section presents a description of the fuel pin cell, fuel, blanket assembly and full core. The fuel pin cell, assembly, and the whole core are presented in Fig: 3.7, Fig: 3.8, and Fig: 3.9 respectively. A single fuel pin cell analysis with specular reflective

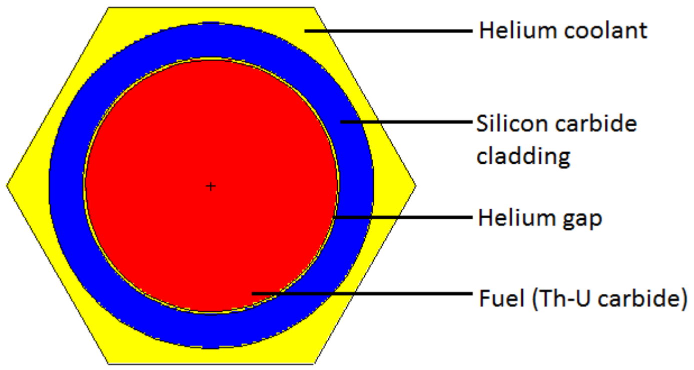


Figure 3.7: Single fuel pin cell with specular reflective boundary conditions. Reprinted with permission from [39].

conditions for all external boundaries is conducted to determine the behavior of the

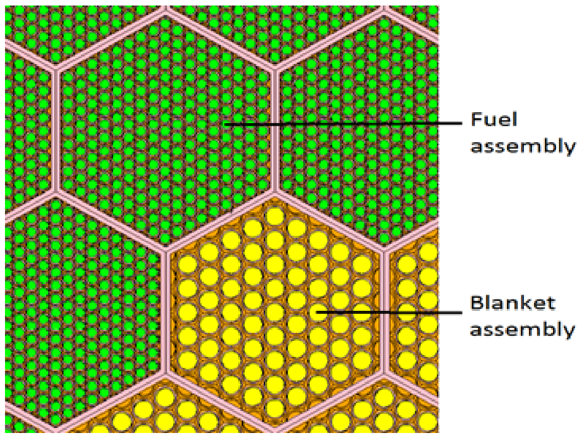


Figure 3.8: GCFBR Assembly. Reprinted with permission from [39].

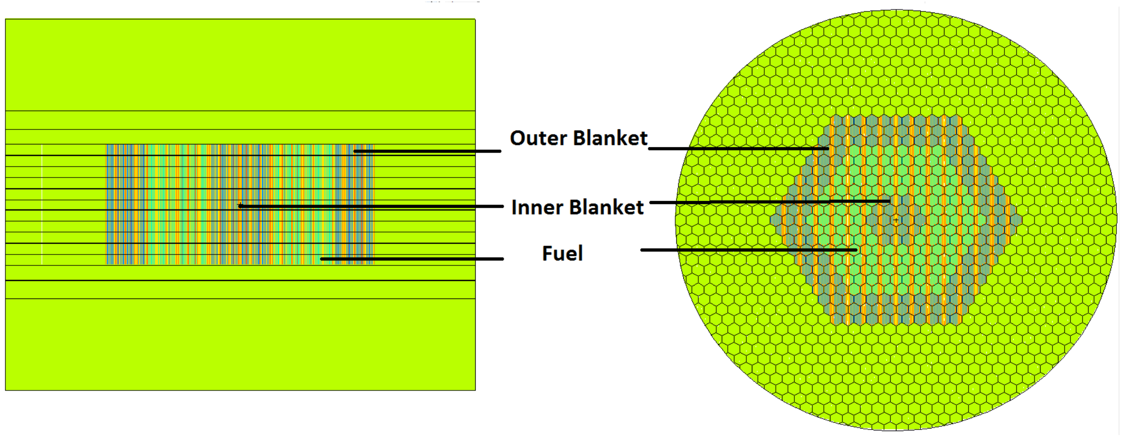


Figure 3.9: Axial and radial view of the whole core.

parameters: infinite neutron multiplication factor of the fuel pin cell configuration, k_∞. To obtain a controlled nuclear fission chain reaction with breeding capability the following objectives need to be met: higher value for $k_\infty > 1$. The parameters analyzed in the fuel pin cell module are, the radius of the fuel pin in the fuel element (r_f), and the enrichment of U-233 in Th-U fuel ([39]). It is to be noted that, $k_\infty > 1$ depends on the $\frac{p}{D}$ ratio. In this case a constant value for the pitch is used, hence, the diameter varies due to varying $\frac{p}{D}$ ratio. Fig: 3.7 shows a graphical view of the single fuel pin cell used for the analysis.

For the whole core analysis, the configuration of the core of an existing gas cooled fast breeder reactor design [39] is used. However, for simplicity, the control rods, and the axial blankets have been ignored in the design. A detailed description of other components are presented in the paper [39]. The core consists of an array of hexagonal assemblies wherein, an assembly consists of an array of fuel elements in a hexagonal lattice. The assemblies with the fuel elements having fissile material are called as the "fuel" assemblies, and those with fertile material are called as the "blanket" assemblies. The fuel assemblies have fuel elements having a mixture of Th-232 and U-233. The blanket assemblies have light water reactor used fuel. The core consists of internal and external blanket assemblies. The parameters analyzed in this module are: effective neutron multiplication factor, k_{eff}, radial power peaking factor of the core, $F_{\text{PF,rad}}$, and axial power peaking factor of the core $F_{\text{PF,ax}}$. The total mass of the fuel (fissile+fertile) material is kept constant. Therefore, when the radius of the fuel element is varied, the height of the core (L_{FC}) is affected, and due to having a constant density, the total mass of the fuel remains constant. For a safe operation in terms of preventing meltdown of the fuel rod, peaking factors play an important role. The peaking factor is the ratio between maximum local energy depositions to the average energy deposition in the reactor core. It is assumed that the external blanket is not part of the reactor core while calculating the radial peaking factors. In the whole core analysis a constant value for the, power, and positioning of blankets is assumed. Fig: 3.9 presents a radial and an axial view of the whole core. The flux spectrum is given by Fig: 3.10, Following table presents a summary of the design characteristics of the reactor model,

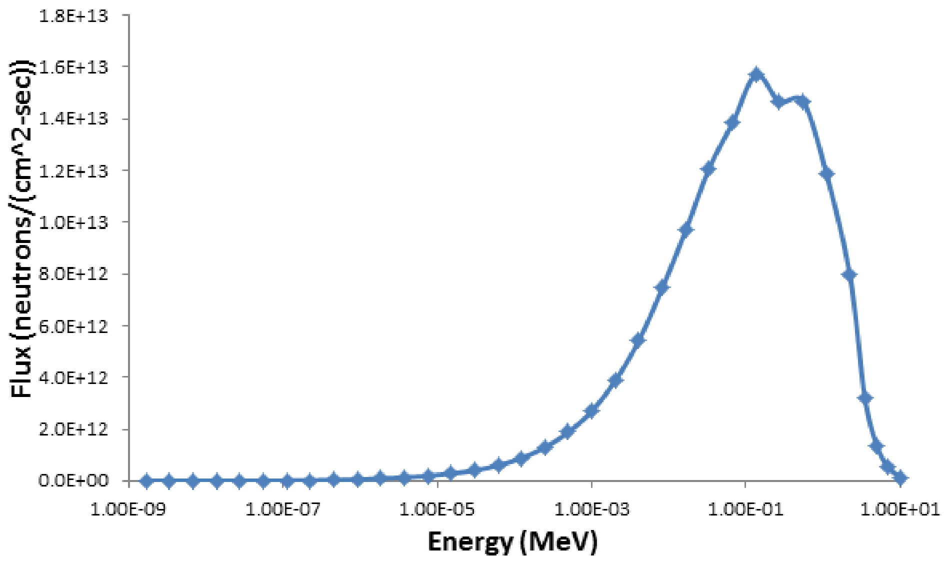


Figure 3.10: Energy spectrum in GCFBR. Reprinted with permission from [39].

Parameter	**Value**	**Units**
Number of pins per blanket assembly	61	
Number of pins per fuel assembly	217	
Reactor power	445	MW_{th}
Mass of initial content of fissile material	2008	kg
Height of core	2.3	m
Outer radius of core	2.6	m

Table 3.7: GCFBR design summary.

3.3.1.2 Thermal Hydraulics

A standard analysis of the transfer of heat across the fuel gap, cladding and bulk coolant is performed for the core. Since the primary focus is the implementation of GA in the multiple module domain coupled with the regression analysis, a rigorous thermal hydraulics and energy transfer analysis has not been performed. For completeness, a simplified model is implemented. In a single phase coolant heat transfer domain, the pressure drop across the length of the active core is the sum of the pressure drop due to friction, form, and elevation is given by,

$$\Delta P = \Delta P_{friction} + \Delta P_{form} + \Delta P_{elevation}. \tag{3.4}$$

For simplicity we assume that the total pressure drop is only due to friction, hence, $\Delta P_{form} = 0$, and $\Delta P_{elevation} = 0$. Therefore, the primary loop pressure drop (ΔP) is given by a simplified equation,

$$\Delta P = \left(\frac{\rho_{FC} \cdot V_{FC}^2}{2}\right) \cdot \left[f_{Darcy-Weisbac} \cdot \frac{L_{FC}}{D_{FC}}\right], \tag{3.5}$$

where

$$V_{FC} = \frac{W}{N_{FC} \cdot \rho FC \cdot A_{FC}}, \tag{3.6}$$

where ρ_{FC} is the density of the coolant, V_{FC} is the velocity of the coolant, $f_{Darcy-Weisbac}$ is the Darcy-Weisbach constant can be estimated as 0.016, D_{FC} is the diameter of the fuel element, N_{FC} is the number of fuel elements, and A_{FC} is the flow area of the coolant. The temperature of the coolant at core outlet (T_{out}) is given by,

$$T_{out} = T_{in} + \frac{Q}{W \cdot C_p}, \tag{3.7}$$

where Q is the total thermal power of the reactor core, and C_p is the specific heat capacity of the coolant. The pumping power of coolant (P_{pump}) is given by,

$$P_{pump} = \frac{\Delta P \cdot A_{FC} \cdot V_{FC}}{\eta_{pump}}, \tag{3.8}$$

where η_{pump} is the pump efficiency. For a safe operation, and to ensure fuel material structural integrity, it is important to compute the maximum radial, and axial fuel temperature. There is a temperature variation radially on the fuel element due to the presence of heterogeneous components: fuel, gap, and clad. The governing equations are,

$$\Delta T_b = \frac{q'_{peak}}{2 \cdot \pi \cdot L_{FC} \cdot (r_F + t_G + t_C)}, \tag{3.9a}$$

$$\Delta T_C = \frac{q'_{peak}}{2 \cdot \pi k_C} \cdot \ln\left(\frac{r_F + t_G + t_C}{r_F + t_G}\right), \tag{3.9b}$$

$$\Delta T_G = \frac{q'_{peak}}{2 \cdot \pi k_G} \cdot \ln\left(\frac{r_F + t_G}{r_F}\right), \tag{3.9c}$$

$$\Delta T_F = \frac{q'_{peak}}{2 \cdot \pi k_F}, \tag{3.9d}$$

where ΔT_F is the temperature drop across the fuel, ΔT_G is the temperature drop across the gap, ΔT_C is the temperature drop across the cladding, ΔT_b is the temperature drop across the bulk coolant, q'_{peak} is the peak linear heat generation rate i.e. linear heat generation rate multiplied by the radial and axial peaking factors, k_C is the thermal conductivity coefficient of the cladding, k_G is the thermal conductivity coefficient of the gap, and k_F is the thermal conductivity coefficient of the fuel. The objectives are to maintain a peak fuel temperature and ΔP within structural integrity limits. The parameters to optimize are T_{in}, and W.

3.3.1.3 Energy Conversion

A Brayton cycle is used to analyze energy conversion of Helium. Efficiency of energy conversion from heat to electricity is important from the economics point of view. A higher efficiency is desired. For simplicity, a simple variant of the Brayton cycle with no regeneration, and reheating is implemented. The optimum pressure ratio ($r_{p,opt}$) is given by,

$$r_{p,opt} = \left(\frac{T_3}{T_1}\right)^{\frac{\gamma}{\gamma-1}}, \tag{3.10}$$

where T_1 is the temperature of coolant at core inlet, T_3 is the temperature of coolant at core outlet, and γ is the heat capacity ratio. The amount of work done by the turbine per unit mass flow rate of the coolant ($\dot{W}_T$) is given by,

$$\dot{W}_T = \eta_T \cdot C_p \cdot T_3 \cdot \left[1 - \frac{1}{{r_p}^{\frac{\gamma-1}{\gamma}}} \right], \tag{3.11}$$

where η_T is the turbine efficiency, and C_p is the specific heat capacity of the coolant. The amount of work done by the compressor per unit mass flow rate of the coolant ($\dot{W}_{CP}$) is given by,

$$\dot{W}_{CP} = \frac{C_p \cdot T_1}{\eta_{CP}} \left[{r_p}^{\frac{\gamma-1}{\gamma}} - 1 \right], \tag{3.12}$$

where η_{CP} is the compressor efficiency. Therefore, the maximum amount of work done ($\dot{W}_{max}$) is given by,

$$\dot{W}_{max} = C_p \cdot T_1 \left[\frac{T_3}{T_1} - {r_p}^{\frac{\gamma-1}{\gamma}} \right]. \tag{3.13}$$

Thermal efficiency (η_{eff}) is given by,

$$\eta_{eff} = \frac{\dot{W}_T - \dot{W}_{CP}}{\dot{W}_{max}}. \tag{3.14}$$

The model of a gas cooled fast breeder reactor (GCFBR) design [39] is used to evaluate the IHGOM. The objective is to optimize a set of parameters in the GCFBR design. The illustrative reactor power system model consists of the following physics based components: heterogeneous neutronics model for criticality, flux, and depletion calculations, basic thermal hydraulics model for heat transfer calculations, and a simple balance-of-plant model for Brayton's cycle calculations. The input parameters given by $\hat{X}$ in (1.38) is a vector of the following design parameters: radius of the fuel element, r_F (RADIUS), enrichment of U-233 in $(U - Th)O_2$ fuel (ENRICH), temperature of the coolant at core inlet, T_{in} (TIN), and flow rate of the coolant, W (W). The global objective function $F\left(\hat{X}\right)$ is a combination of individual objective functions, $f_o\left(\hat{X}\right)$ that maximizes or minimizes a set of QOIs.

3.3.2 Optimization Problem

This section, presents the design parameters, objectives and constraints used in the optimization problem. Table: 3.8 presents the list of the individual objective functions $f_o(X)$ and the corresponding QOIs. In this problem, the number of objectives (O) is six. The objectives are determined such that the reactor stays critical and maintains: a flat power profile, a higher pressure drop in limits, and a high thermal efficiency. There are some of the input samples tracked and discarded due

Label	**QOI**	**Objective**	**Units**
$f_1(X)$	k_∞	Maximize k_∞	
$f_2(X)$	k_{eff}	k_{eff} in limits	
$f_3(X)$	$F_{\text{PF,rad}}$	Minimize $F_{\text{PF,rad}}$	
$f_4(X)$	T_{out}	T_{out} in limits	°C
$f_5(X)$	ΔP	ΔP in limits	Pa
$f_6(X)$	η_{eff}	Maximize η_{eff}	

Table 3.8: Optimization objective functions.

to the nature of the behavior of the physics model e.g. an input sample that makes the reactor sub-critical is discarded from the GA search. A standard approach is to penalize the fitness function, when the search encounters these samples. This is done using constraint functions. Table: 3.9 presents the constraints used in the optimization execution. The penalty is imposed by setting the fitness value to be zero. Table: 3.10 presents the list of the design parameters used in the optimization

Parameter	**Range**	**Units**
$F(X) = 0$	$k_\infty < 1.0$	
$F(X) = 0$	$k_{\text{eff}} < 1.0$	
$F(X) = 0$	$T_{\text{out}} > 600.0$	°C
$F(X) = 0$	$\Delta P > 100.0$	Pa

Table 3.9: Constraints in GCFBR optimization

process,

Parameter	**Range**	**Units**
r_F	$0.2 \rightarrow 0.33$	cm
En	$14 \rightarrow 20$	wt %
T_{in}	$50 \rightarrow 200$	°C
W	$20 \rightarrow 100$	$\frac{kgs}{sec}$

Table 3.10: Input value ranges for the design parameters.

3.3.3 Epsilon Constraint Steps

The ϵ constraint steps consists of several single-obejctive optimization sequences with varied constraints. The steps for the GCFBR problem based on the objectives 3.8 are:

1. Maximize k_∞ by determining the local maximum at several ϵ intervals of other objectives $(f_2 ... f_6)$ used as constraints. The maximum value for k_∞ is called as $k_{\infty,max}$.

2. With $k_\infty = k_{\infty,max}$, maximize k_{eff} upto a limit by determining the local maximum at several ϵ intervals of other objectives $(f_3 ... f_6)$ used as constraints. The maximum value for k_{eff} is called as $k_{\text{eff},max}$.

3. With $k_\infty = k_{\infty,max}, k_{\text{eff}} = k_{\text{eff},max}$, minimize $F_{\text{PF,rad}}$ by determining the local minimum at several ϵ intervals of other objectives (f_4, f_5, f_6) used as constraints. The minimum value for $F_{\text{PF,rad}}$ is called as $F_{\text{PF,rad,min}}$.

4. With $k_\infty = k_{\infty,max}, k_{\text{eff}} = k_{\text{eff},max}, F_{\text{PF,rad}} = F_{\text{PF,rad,min}}$, maximize T_{out} upto a limit by determining the local maximum at several ϵ intervals of other objectives (f_5, f_6) used as constraints. The maximum value for T_{out} is called as $T_{\text{out,max}}$.

5. With $k_\infty = k_{\infty,max}, k_{\text{eff}} = k_{\text{eff},max}, F_{\text{PF,rad}} = F_{\text{PF,rad,min}}$ and $T_{\text{out}} = T_{\text{out,max}}$, maximize ΔP upto a limit by determining the local maximum at several ϵ intervals of f_6 used as constraints. The maximum value for ΔP is called as ΔP_{max}.

6. With $k_\infty = k_{\infty,max}, k_{\text{eff}} = k_{\text{eff},max}, F_{\text{PF,rad}} = F_{\text{PF,rad,min}}, T_{\text{out}} = T_{\text{out,max}}$ and $\Delta P = \Delta P_{\text{max}}$, maximize η_{eff}.

3.3.4 Iterative Method Parameters

Table: 3.11 presents the values used in the HYBGASA implementation and execution. The parameters include the kernel functions, tuning parameters in GA, ϵ constraint parameters and GPM hyper parameters. The constraint limits for the QOIs in the objective functions for the ϵ constraint method is presented in Table: 3.12.

	Parameter	**Values**
GA	Number of chromosomes	200
	η	0.05
	ν_{mut}	0.5
	c_k	0.25
	τ_{IF}	1.0E-04
	$\tau_{relDiff}$	1.0E-02
	τ_{NBest}	10
ϵ constraint	intervals	11
	$N_{initial}$	80
Outer	$N_{re-train}$	40
	retrain space	0.5
	convergence threshold	1.0E-06

Table 3.11: Optimization parameters

Parameter	**Range**	**Units**
k_∞	$1.0 - 1.5$	
k_{eff}	$1.006 - 1.014$	
$F_{\text{PF,rad}}$	$1.0 - 2.0$	
T_{max}	$400.0 - 600.0$	°C
ΔP	$60.0 - 100.0$	Pa
η_{eff}	$40.0 - 60.0$	%

Table 3.12: Constraint limits in the ϵ constraint implementation

3.3.5 Results

3.3.5.1 Parameter Search and Sensitivity Studies

The first step involves an extensive search for the values of the parameters used in the regression model and the ϵ-constraint method. Search studies have been done to determine the following parameters/functions,

1. Kernel function for GPM

2. GPM hyper-parameters

3. ϵ intervals

3.3.5.2 GPM Kernel Function and Hyper Parameter Search

A grid based search is performed to determine the kernel function and the hyper parameter that performs the best prediction for a QOI in every outer iteration. The grid search is given in Table: 3.13, Based on the combinations of the above

Kernel Function	θ_d **grid**
Absolute exponential	0.05-1.0 (Intervals-0.05)
Squared exponential	0.05-1.0 (Intervals-0.05)
Cubic	0.05-1.0 (Intervals-0.05)
Linear	0.05-1.0 (Intervals-0.05)

Table 3.13: Kernel function and hyper-parameter search grid

grid a search is performed and the results of the search at the end of the first outer iteration is given in Fig: 3.11, The first 20 points on the x-axis represent the "absolute exponential", the second 20 points represent the "squared exponential", the third 20 represent "cubic" and the last 20 represent the "linear" kernel function. For each function, a set of twenty values for θ_d is analyzed, and the MPE for each of these are shown on the y-axis. θ_d ranges from 0.05 to 1.0 with an interval of 0.05. The colored lines represent a the QOIs. It is evident that different QOIs have their own choice of the kernel function that gives the best predictions. It is assumed that the MPE in the predictions from the GPMEM built using a function and θ_d is directly proportional to the quality of the predictions i.e. a function-θ_d pair that prouces the least MPE is the best prediction model. For QOI-1, "squared exponential" kernel performs the best as compared to other kernel functions, while for QOI-3, "cubic" has the best predictions. The next step is to determine the exact value of θ_d that gives the best

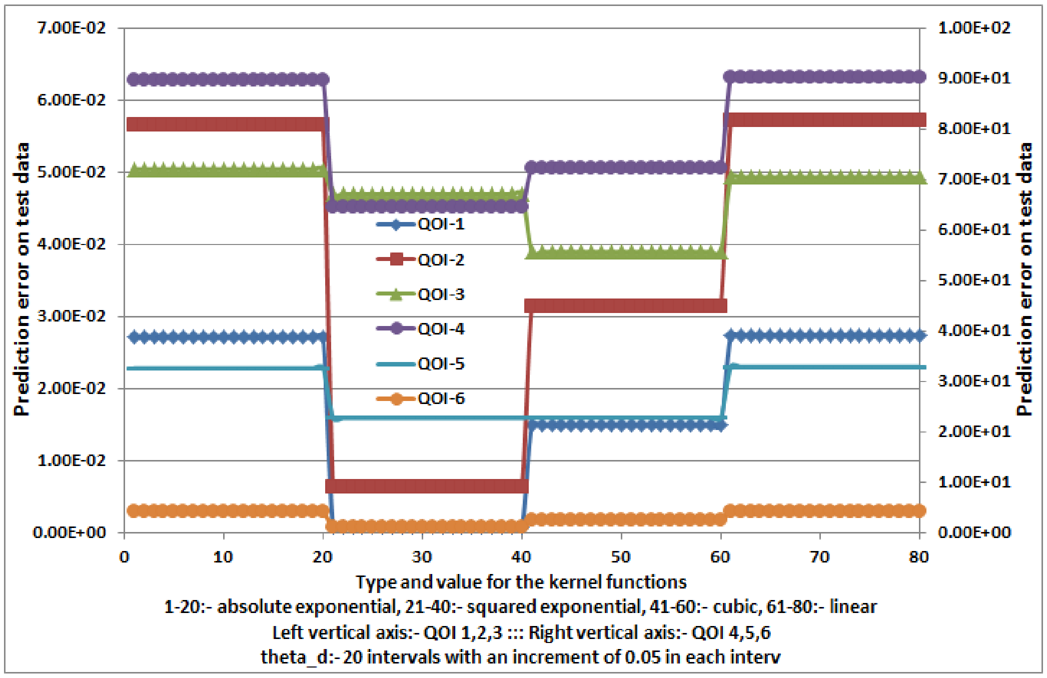


Figure 3.11: Test prediction error versus kernel function and hyper parameter for the QOIs at the beginning of the first outer iteration

predictions for a QOI. The choice of θ_d along with the best kernel function for each QOI in outer iteration-1 is given in Fig: 3.12 and Table: 3.14. Each of the sub-plots

Outer Iteration	**QOI**	**Kernel Function**	θ_d
1	1	Squared exponential	0.8
	2	Squared exponential	0.8
	3	Cubic	0.6
	4	Squared exponential	0.15
	5	Cubic	0.5
	6	Squared exponential	0.65

Table 3.14: Kernel function and hyper-parameter results

in Fig: 3.12 represents the best performing kernel function for each of the QOIs.

The color of the subplots are consistent with the color of that specific QOI in Fig: 3.11. It is to be noted that the kernel functions and the hyper parameter depends on the training data, and therefore the choice is made at the beginning of every outer iteration. From the analysis of the outer iterations it is observed that the choice of

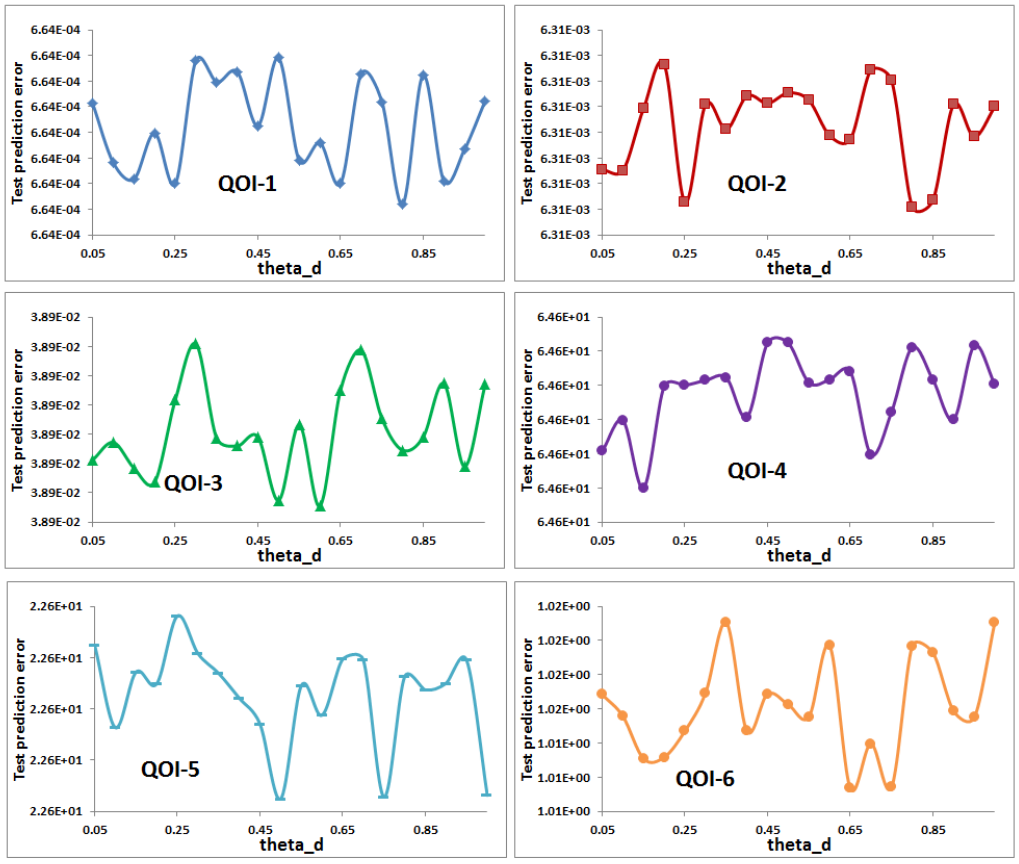


Figure 3.12: Kernel function and hyper parameter versus test prediction error for a set of QOIs at the beginning of first outer iteration. The θ_d plots correspond to the best performing kernel function for each of the QOIs.

the kernel function that gives the best prediction for a specific QOI does not depend on the outer iteration, however, the choice of θ_d depends on the outer iterations.

This can be inferred from the fact that the regression surface is significantly affected because of re-training the model. The results for other outer iterations are shown in the Appendix.

3.3.5.3 The ϵ Intervals

A sensitivity study is performed by changing the number of ϵ intervals and investigating the change in the fitness values. The sensitivity study is performed during the beginning of every outer iteration when the training sample size is updated. A threshold value of $1.0E-04$ is used to determine the number of intervals i.e. the number of intervals corresponding to a relative change in the fitness value of $\leq$ threshold is the final number of intervals in that outer iteration. From Fig: 3.13 the number

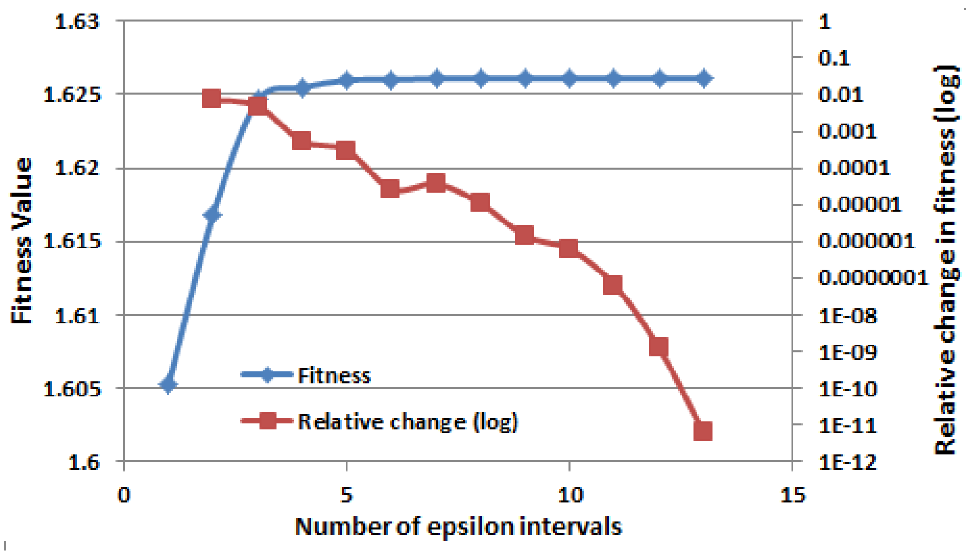


Figure 3.13: Fitness value vs ϵ intervals for the first outer iteration

of intervals is chosen as 5. It is not known whether the defined system has a convex objective space or not, therefore the ϵ sensitivity study helps to investigate if there is concavity in the space. A convergence of the fitness value to a saturated point with the change in ϵ intervals indicate that the concavities have been taken care of in the intervals. Figures 3.14 and 3.14 show the convergence to a solution along the

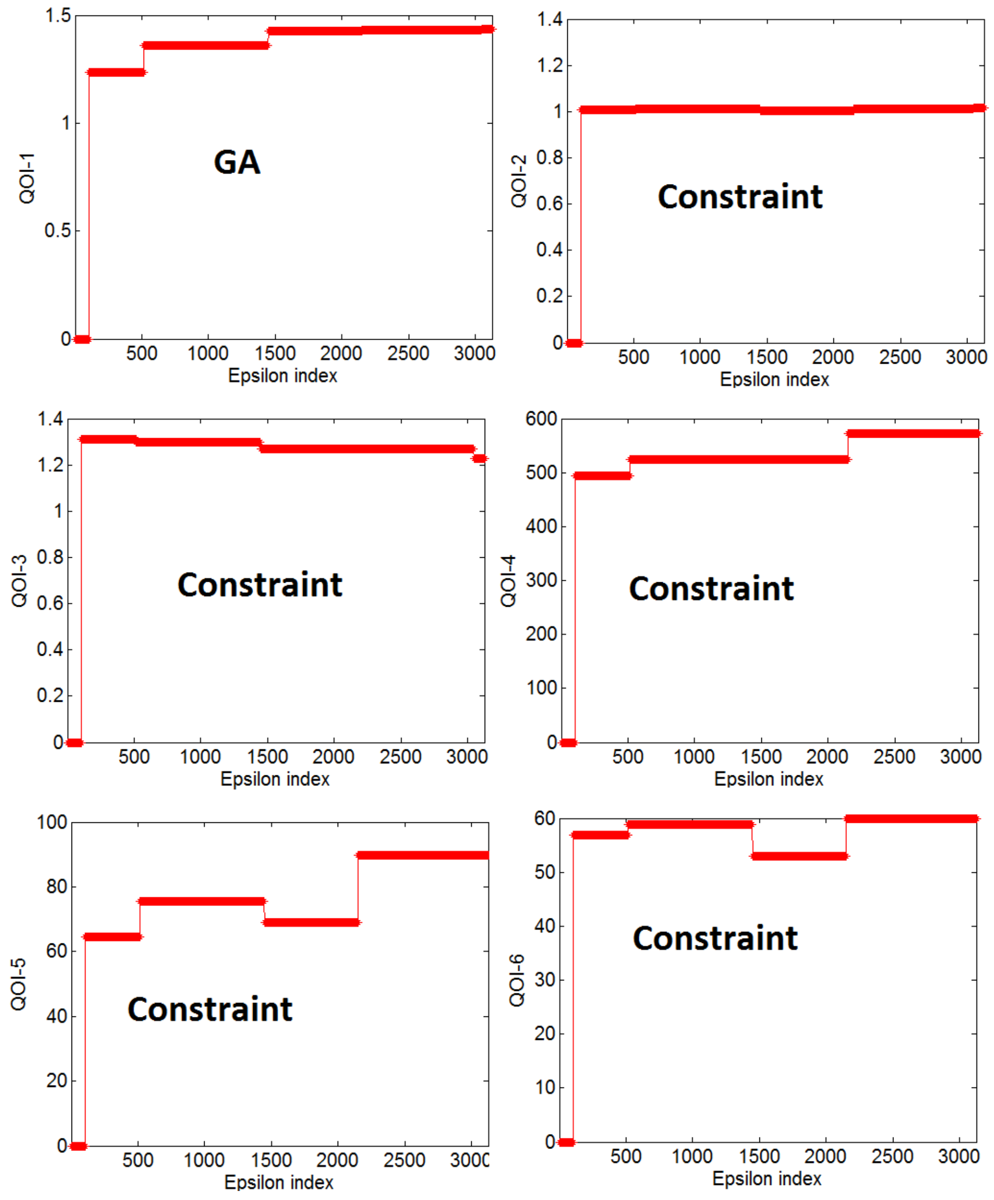


Figure 3.14: Convergence to solution for the epsilon intervals, Objective:1

epsilon intervals.

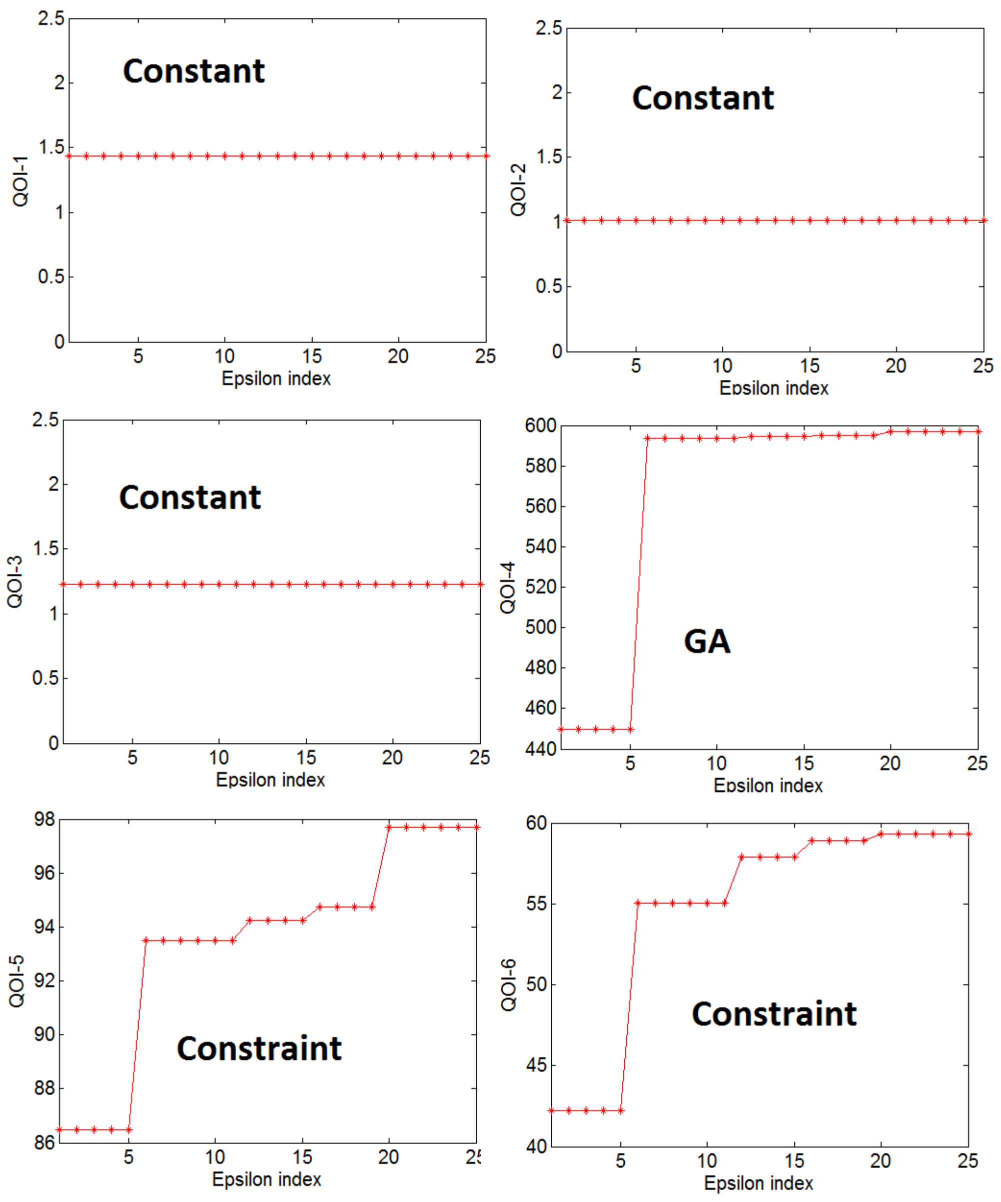


Figure 3.15: Convergence to solution for the epsilon intervals, Objective:4

3.3.5.4 Genetic Algorithms

The convergence of the design parameters during outer iteration:9 (last iteration) for the first objective is shown in Fig: 3.16. The best solution out of all the chromosomes in a particular generation is saved and its search direction across several generations is presented. It is difficult to present all the objectives and their convergence based on the generations for all the epsilon intervals, therefore the problem is run with a single epsilon interval and the convergence criteria is presented. Figs:

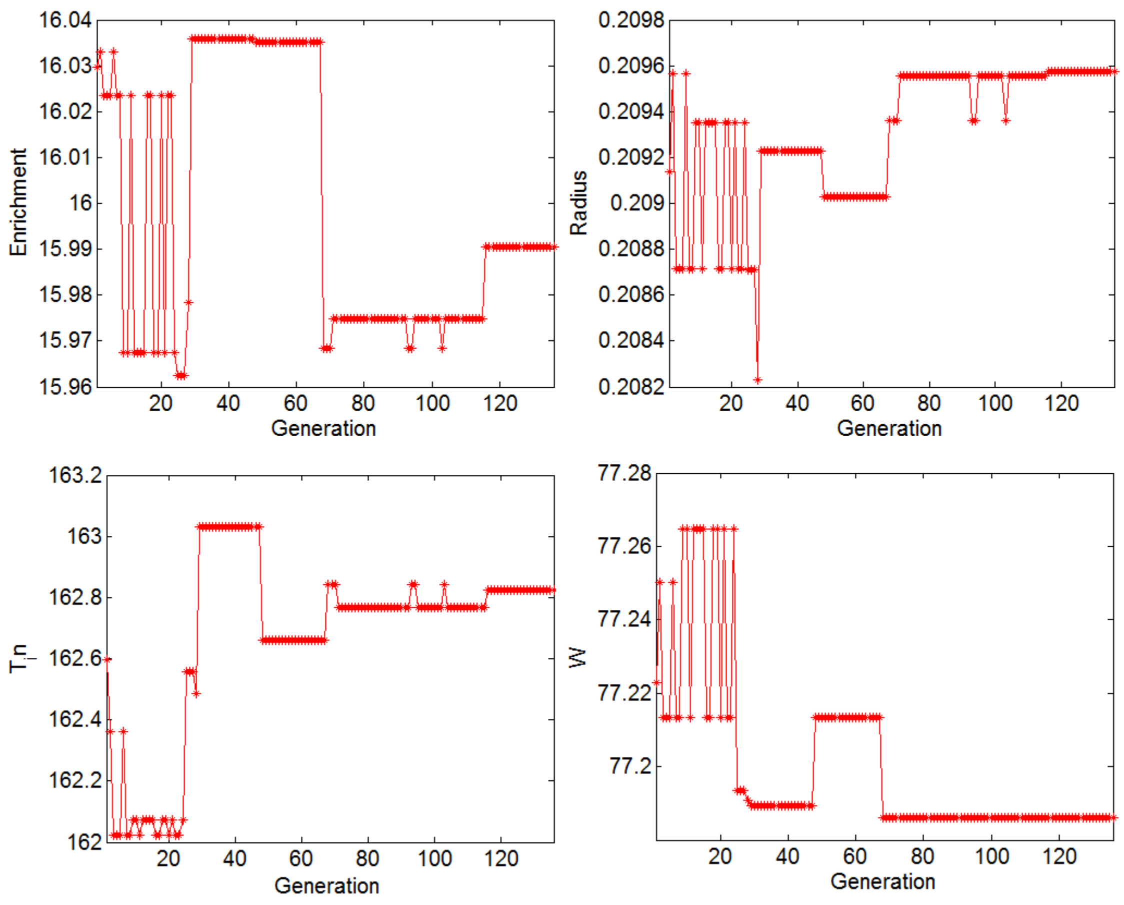


Figure 3.16: Values for the design parameters versus generations in outer iteration:9 for Objective:1

3.17 - 3.22 shows the convergence of each of the QOIs during the outer iteration:9 with a single epsilon interval. The stopping criteria and the search towards

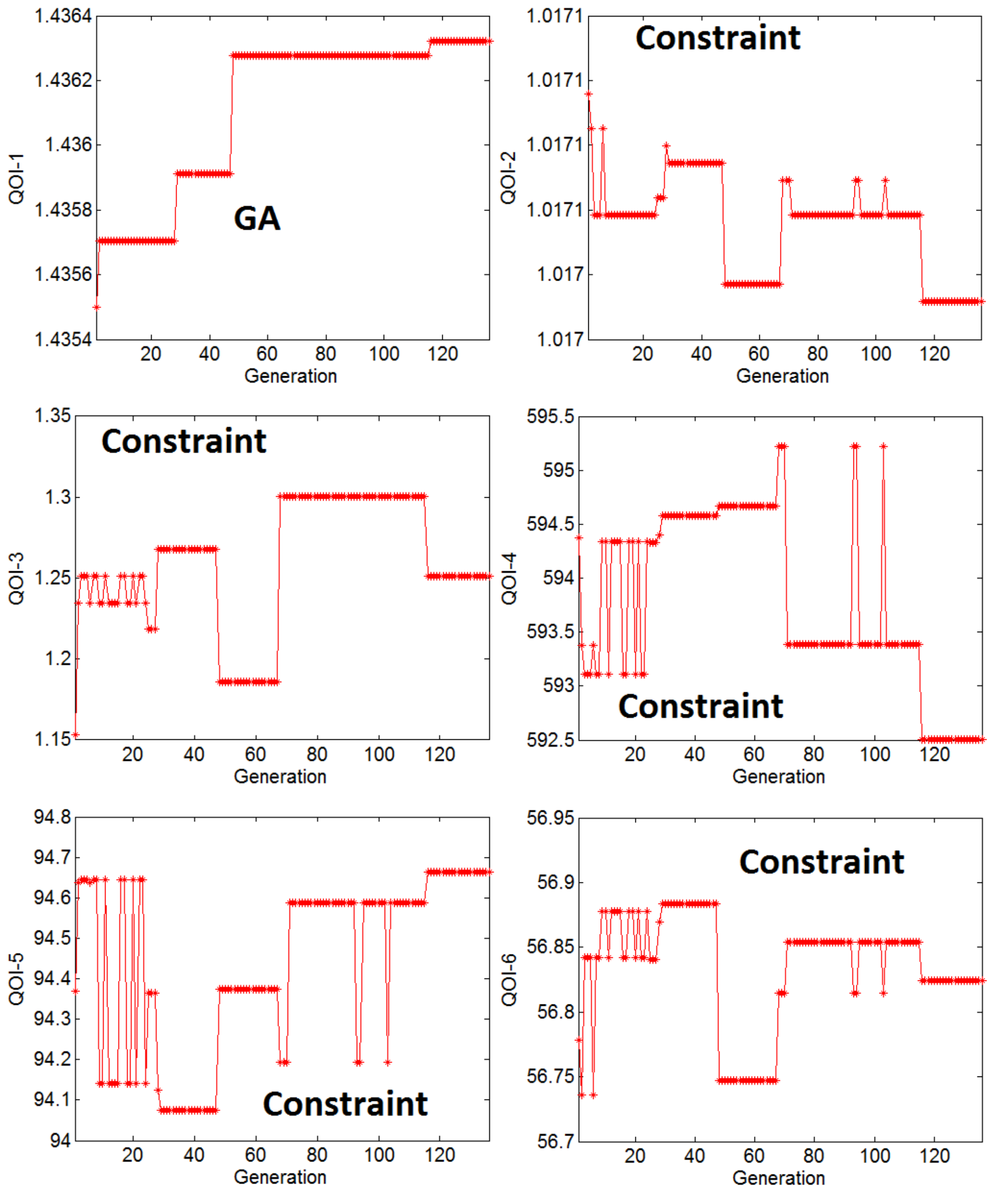


Figure 3.17: Convergence of GA for the QOIs in the last outer iteration with one epsilon interval for Objective:1

GA convergence in the last outer iteration is presented. Fig: 3.23 shows the convergence of the mean and the best solution of all the chromosomes in each generation is presented. Upon convergence of GA, the solution of all the chromosomes move

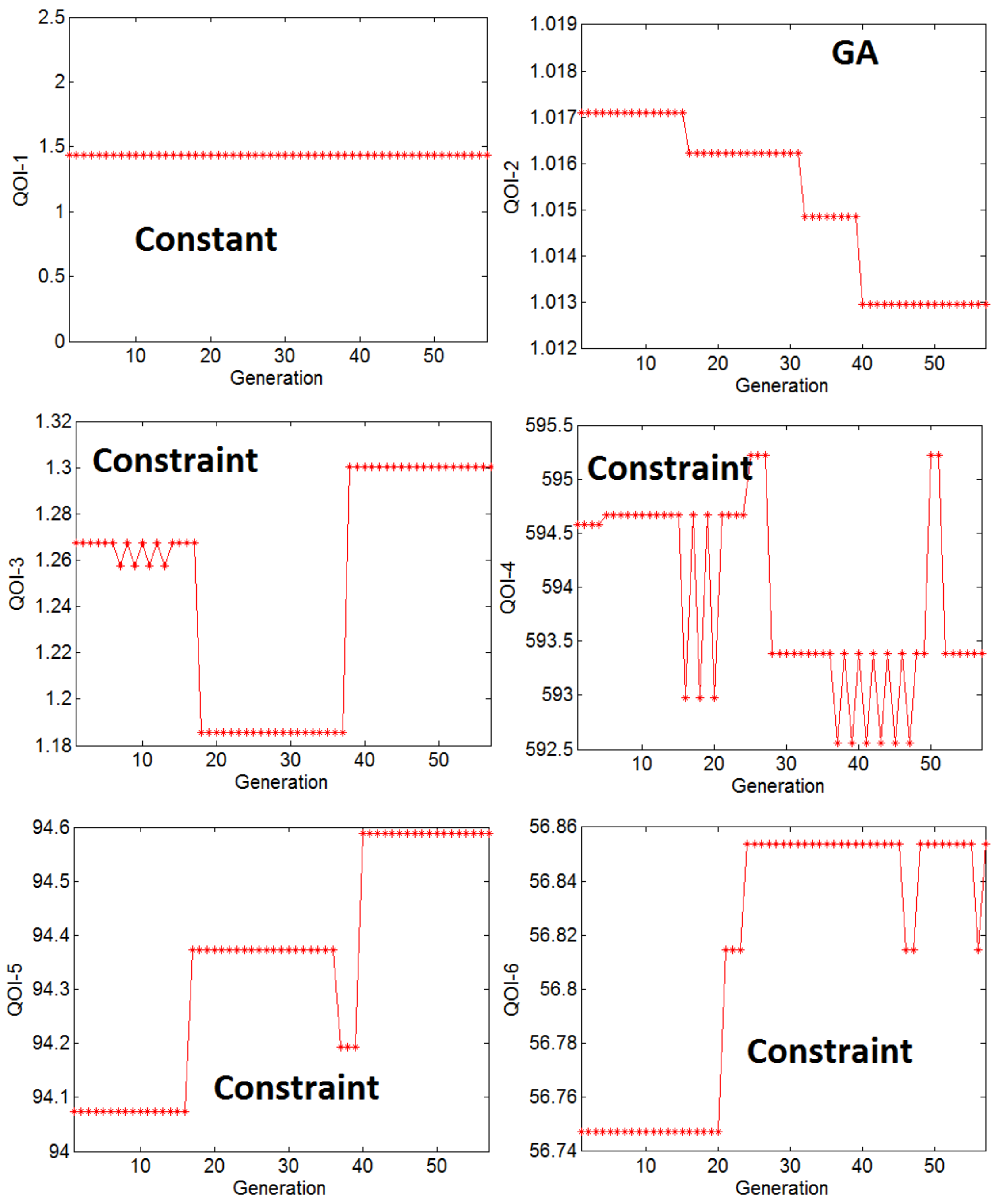


Figure 3.18: Convergence of GA for the QOIs in the last outer iteration with one epsilon interval for Objective:2

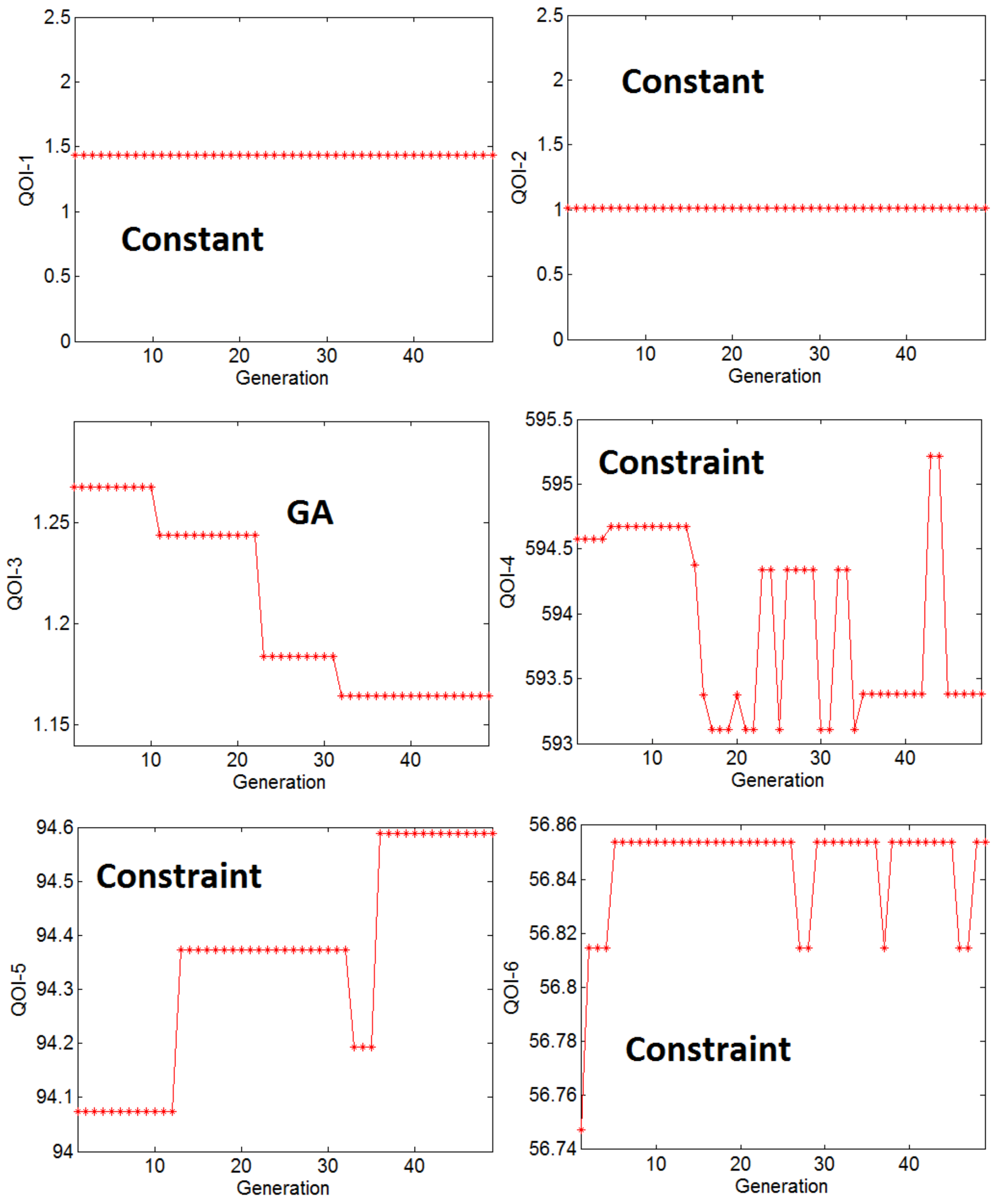


Figure 3.19: Convergence of GA for the QOIs in the last outer iteration with one epsilon interval for Objective:3

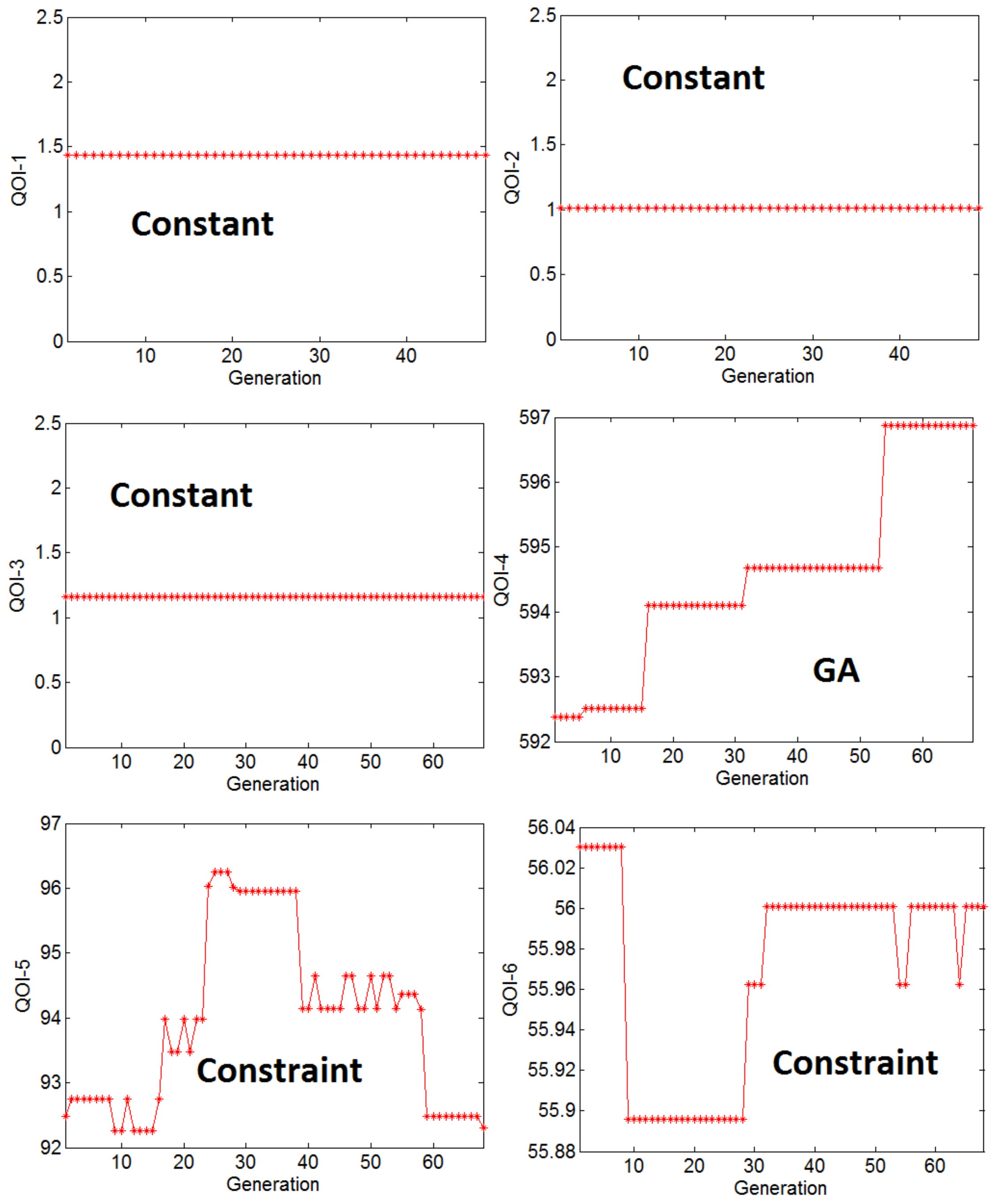


Figure 3.20: Convergence of GA for the QOIs in the last outer iteration with one epsilon interval for Objective:4

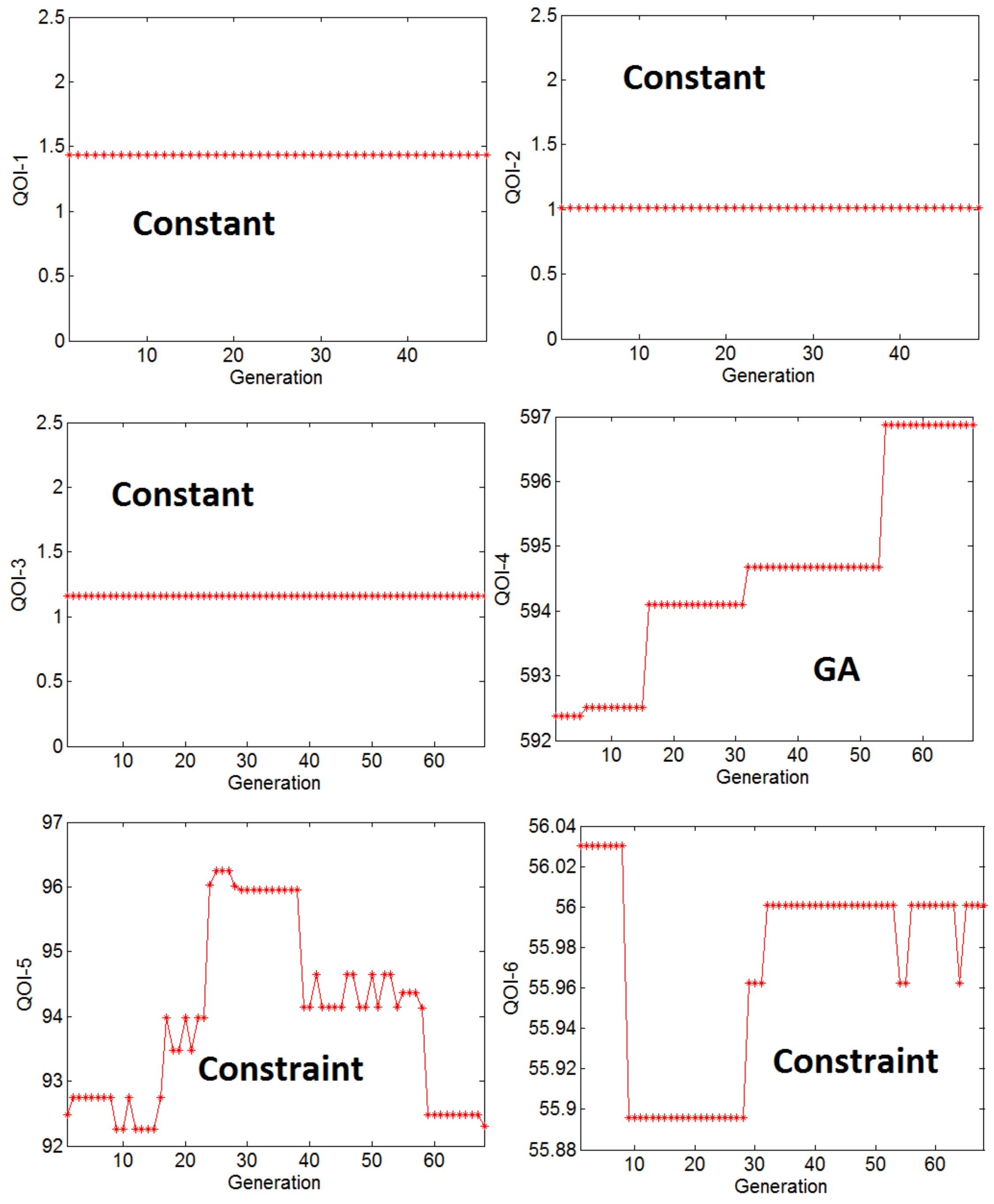


Figure 3.21: Convergence of GA for the QOIs in the last outer iteration with one epsilon interval for Objective:5

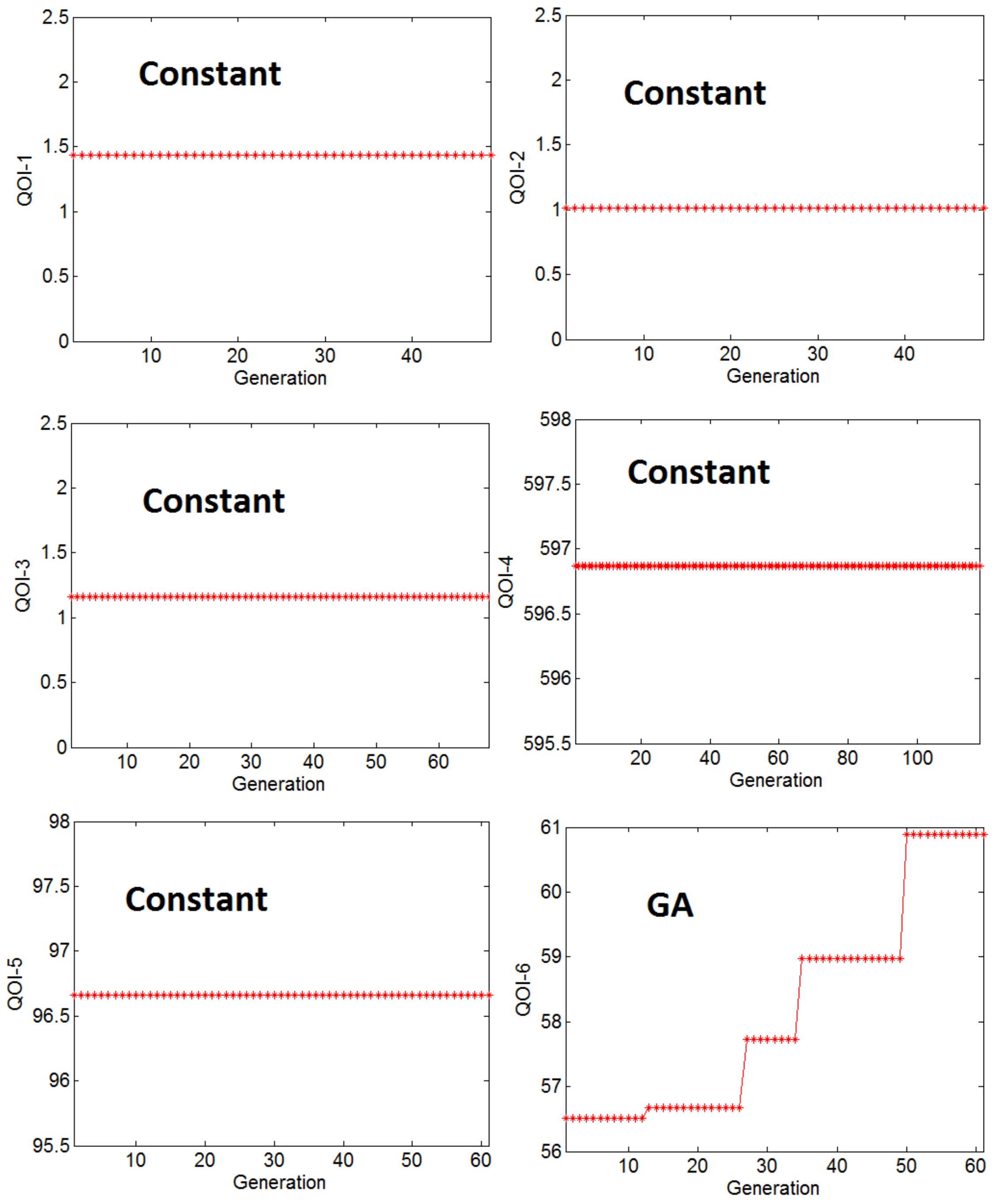


Figure 3.22: Convergence of GA for the QOIs in the last outer iteration with one epsilon interval for Objective:6

towards the optimal solution, and therefore the mean approaches a saturated value. One of the stopping criteria is to track the generations when the relative difference goes below 0.01. The improvement in fitness and the convergence of the best fitness

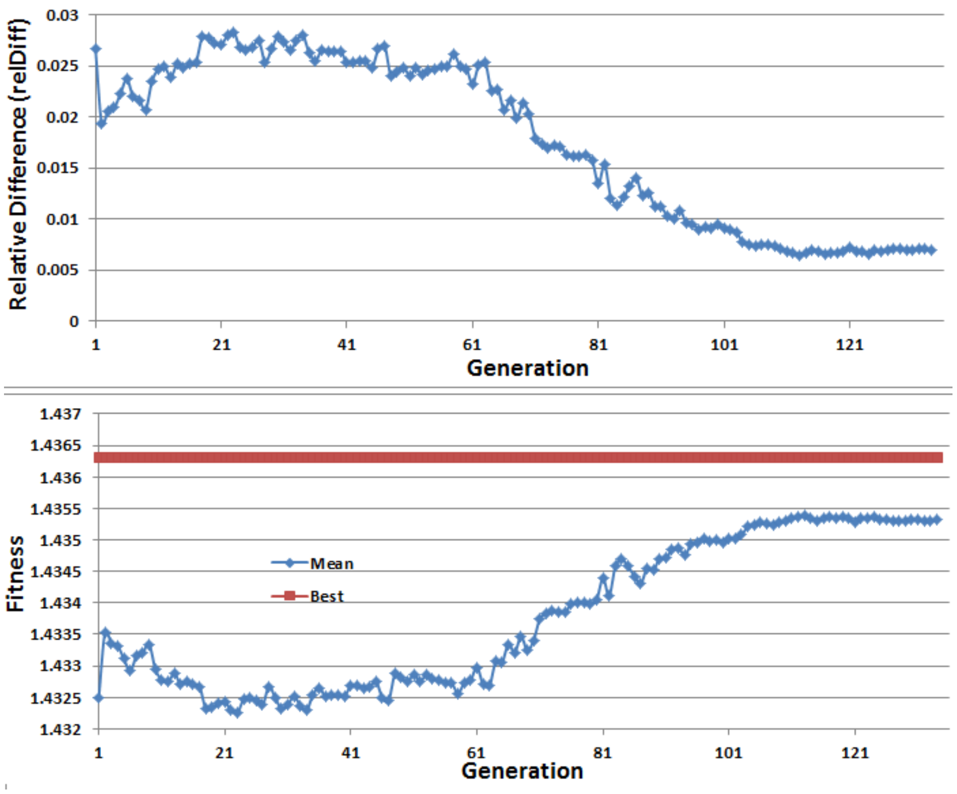


Figure 3.23: Stopping criteria (relative difference between mean and best fitness) and convergence of GA for first outer iteration

is considered as a stopping criteria along with the relative difference criteria. Fig: 3.24 shows the best fitness obtained with generation and the improvement in fitness. There is not significant relationship between an outer iteration and the number of GA generation in that outer iteration.

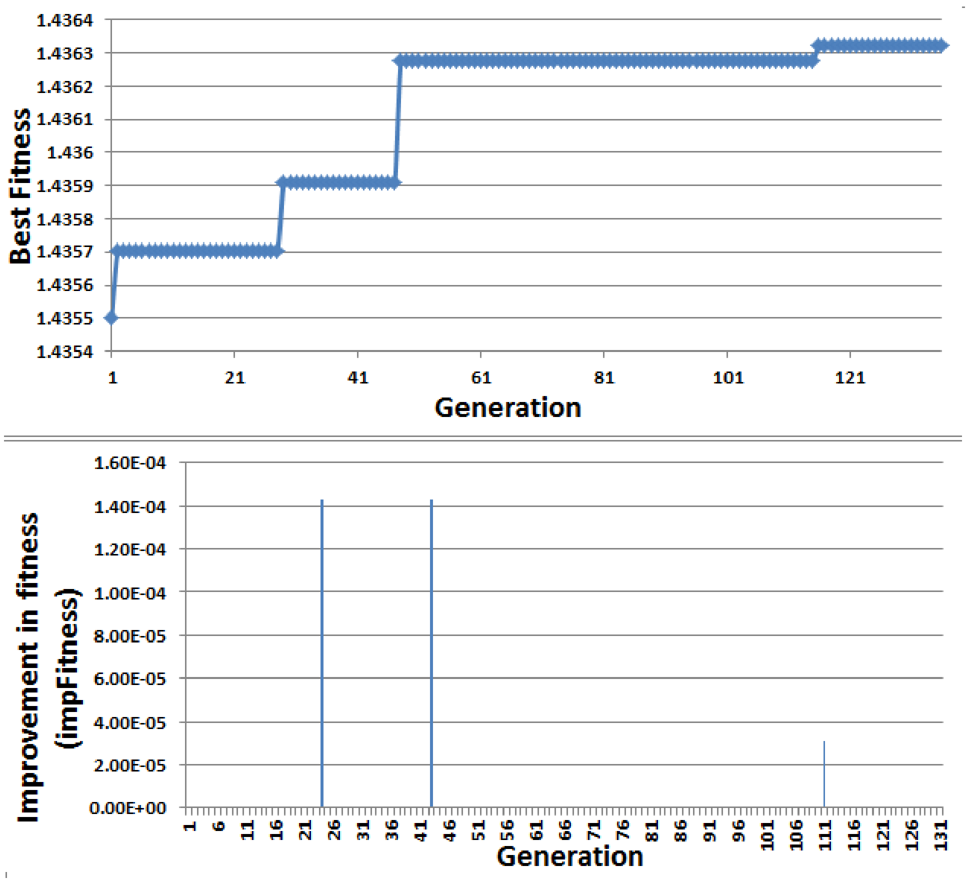


Figure 3.24: Stopping criteria (best fitness and improvement in fitness) and convergence of GA for Outer iteration : 1

3.3.5.5 Outer Iterations

The IHGOM method converged after nine iterations. With an initial sample set of 80, and re-sampling with 40 samples, nine outer iterations involved an execution of a total of 400 samples. Table: 3.15 shows the Pareto optimal values for the design parameters at the end of the convergence of the outer iterations. Table: 3.16 shows the converged values for the QOIs after the end of all the outer iterations. This table contains the actual values and the relative error in predictions obtained from the execution of the actual system. Fig: 3.25 shows the relative error between

Parameter	**Values**
R	$2.01957E-01$
En	$1.6121E+01$
T_{in}	$1.6482E+02$
W	$5.7486E+01$

Table 3.15: A Pareto optimal solution after the convergence of the outer iterations

Parameter	**QOI Predictions**	**QOI Actual values**	$\epsilon_{\text{predErr}}$
$QOI-1$	1.436321339	1.436371535	$3.4946E-05$
$QOI-2$	1.013832	1.013966188	$1.3223E-04$
$QOI-3$	1.227564	1.226349769	$9.9017E-04$
$QOI-4$	596.8746976	596.5838688	$4.8760E-04$
$QOI-5$	97.68926	97.71031716	$2.1551E-04$
$QOI-6$	59.2959929	59.86035672	$3.8018E-04$

Table 3.16: A Pareto optimal solution - QOIs after the convergence of the outer iterations

the predicted value and the actual value of the Pareto optimal solution obtained after the convergence of the outer iterations is also shown. The outer iterations are converged after the MPE error becomes less than $1.0E-05$. This does not mean that the relative error between predicted and actual value is also less than $1.0E-05$. However, the MPE error and $\epsilon_{\text{relDiff}}$ follow the outer iteration together. Fig: 3.26 shows the comparison between the iterative search method and the brute force search for an optimum solution. In the brute force search, the design space is sampled normally using the LHS method. A weighted approach with all the QOIs having equal weights are used to calculate the fitness and compare with the solution obtained from the iterative method. In the figure, the horizontal red line is the fitness of the solution obtained from the iterative method, and the blue bars represent fitness from the solutions obtained using brute force search. It is evident that the iterative

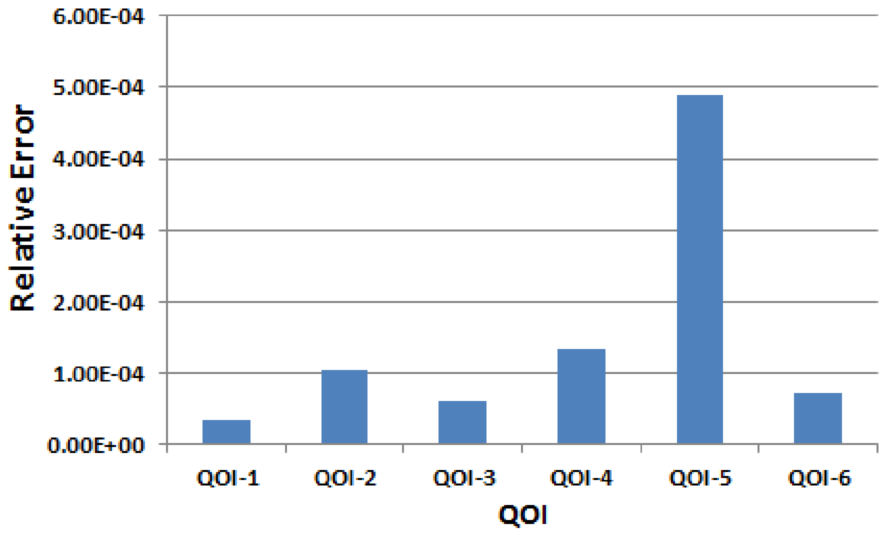


Figure 3.25: Relative error in the QOIs between predicted values versus actual values

convergence had a total of 400 executions of the simulation and the results are better than 3200 brute force executions. This is the speed generated due to the proposed method. Table: 3.17 and Fig: 3.27 show the convergence of the outer iteration for

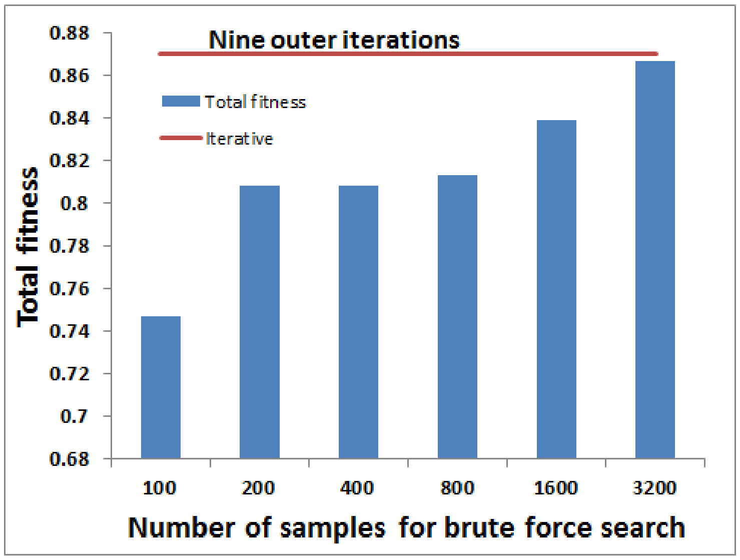


Figure 3.26: Comparison between iterative method and brute force search

QOI-1. The table and the figure compare the test MPE reduction between local re-sampling and global re-sampling. This shows that local re-sampling is more effective as compared to global re-sampling in re-training the GPM for predictions. Fig: 3.28

Outer iteration	**Local re-sampling**	**Global re-sampling**
1	$1.0000E+00$	$1.0000E+00$
2	$1.1554E-01$	$4.3159E-01$
3	$3.3667E-03$	$1.8507E-01$
4	$1.7903E-03$	$1.2234E-01$
5	$1.3037E-03$	$9.3376E-02$
6	$1.9215E-04$	$7.2860E-02$
7	$1.2937E-04$	$6.3565E-02$
8	$4.4935E-06$	$2.0145E-02$
9	$1.4935E-07$	$5.3951E-02$

Table 3.17: Test predictions error in local re-sampling and global re-sampling versus outer iteration for QOI-1

shows the re-sampling space of the design parameters versus outer iteration. With the outer iteration, the difference between the lower and the upper bounds for all the design parameters decreases.

3.3.5.6 Discussion

The novelty of the method is described in the following discussion,

3.3.5.6.1 Speed: A Pareto optimal solution is obtained in 400 executions of the system as compared to 3200 brute force executions. This shows that the developed method would reach to a Pareto optimal solution with a significantly lower number of executions of the system. The comparison between local re-sampling and global re-sampling (Fig: 3.27) shows that our method which used local re-sampling reduces

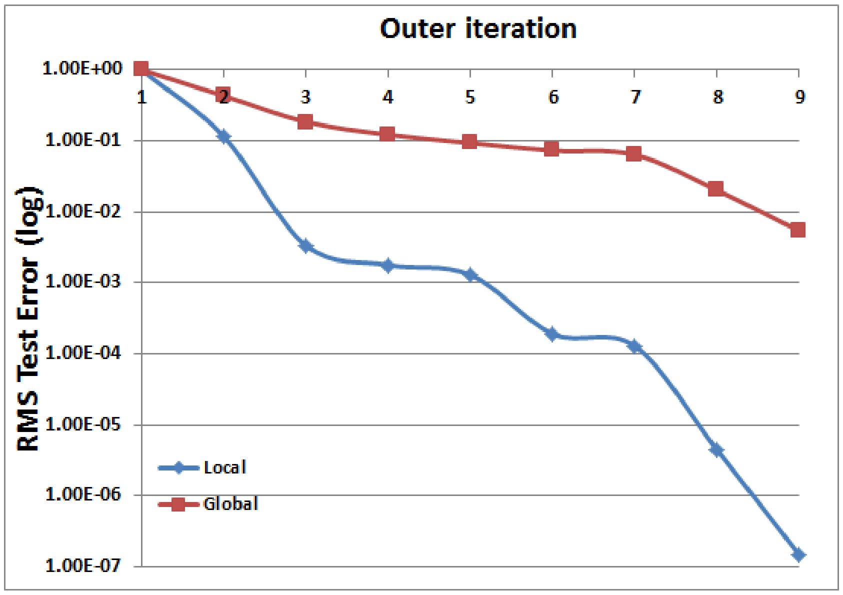


Figure 3.27: Test predictions error in local re-sampling and global re-sampling versus outer iteration for QOI-1

the MPE at a significant speed as compared to the traditional global re-sampling process.

3.3.5.6.2 Accuracy: The accuracy of the method is demonstrated based on the accuracy of the predictions at the end of the outer iterations. The accuracy is measured in terms of the prediction error i.e. how far the predictions for the QOIs are from the values obtained from the experiment/simulation. This is shown in Table: 3.16 and Fig: 3.25.

3.3.5.6.3 Search Efficiency: The optimization capability of the developed method is shown by comparing the solution obtained for the QOIs (Table: 3.16) and the optimization objectives and the constraints, Table: 3.8 and Table: 3.9. Comparing to the objectives and the constraints of the problem,

- As desired based on the objective functions, a higher value for k_∞ of 1.436321339 is obtained.

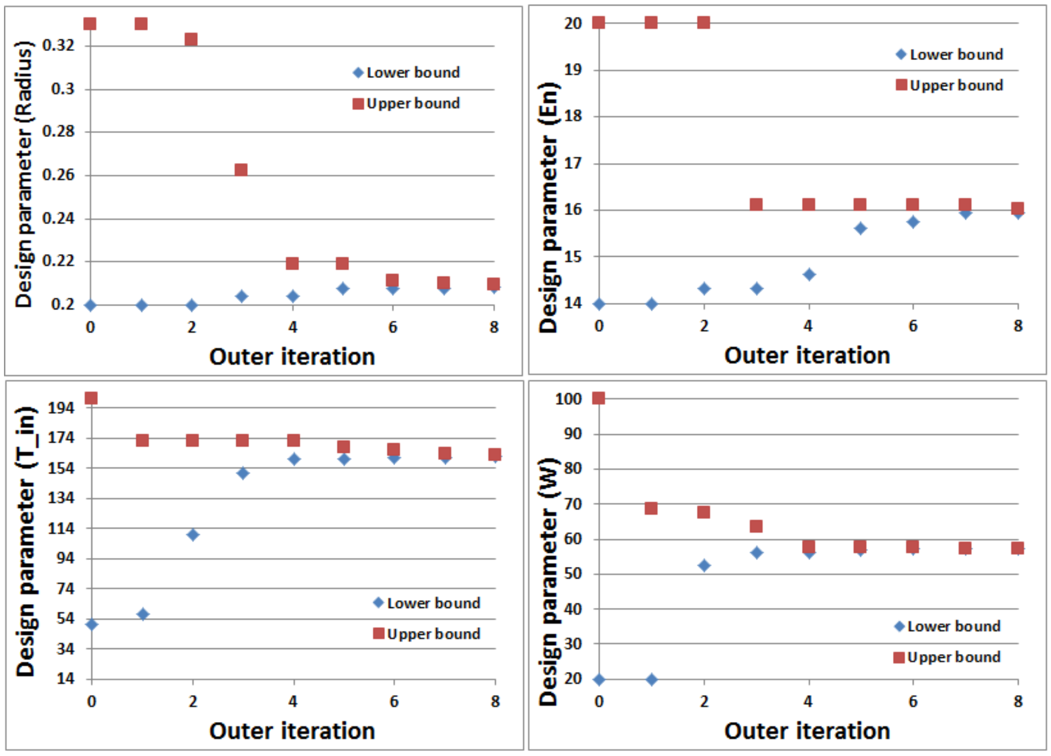


Figure 3.28: The design parameter bounds versus outer iteration

- A value for k_{eff} calculated by the system is 1.017031837 which is closer to 1.01 is obtained.
- A considerably acceptable value for the radial peaking factor, 1.251211832 is obtained.
- The objectives and the constraints desired the value for T_{out} to be maximum and close to 600, and the value for T_{out} obtained by the developed method is 592.5046976.
- The objectives and the constraints desired the value for Δ_P to be maximum and close to 100, and the value for Δ_P obtained by the developed method is 94.6638719.

- A desired maximum value of 56.82436929 is obtained for the thermal efficiency.

The objectives related to the criticality, radial peaking factor, temperature of coolant at the core outlet, pressure drop across the core, and the thermal efficiency have been met subject to the defined constraints and the limits of the design parameters. The optimal solution for the problem is not known, therefore it is not possible to say that the optimal solution has been determined. However, a qualitative approach towards the optimality of the solution show that a good solution based on the defined objectives and the constraints are obtained. The search efficiency has been measured in terms of how efficient is the obtained optimal solution based on the objectives and the constraints. in the above comparison, a discussion on the proximity of the solution QOIs to the objectives is presented.

3.4 Pressurized Water Reactor Neutronics Optimization

This section presents the design specification, design parameters, objectives, and constraints in the optimization of the neutronics aspects of a pressurized water reactor model (PWR). The model is similar to the AP1000 reactor model developed by Westinghouse [60]. The design is developed from the specifications submitted to the Nuclear Regulatory Commission. Henceforth the reactor will be referred to as the Modified AP1000 model (MAP1000). For simplicity, the reactor model does not have control rods. The objective is to determine the optimum values for the fuel enrichment, and power in order to obtain a sustainable and safe design for the core. Sustainability is measured in terms of the burn up, and safety is measured in terms of the peaking factor and the criticality at the beginning of core life. The tasks are to start with the base model presented in literature, and search for a better Pareto optimal solution based on the priority of the objectives. Following the optimization search, a couple of solutions have been identified that have a better set of parameters compared to the base model. It is to be noted that a rigorous optimization search can be performed by tuning the optimization parameters and designing the objectives and constraints in a more realistic manner. This rigorous search is beyond the scope of this dissertation, however the examples presented here are an illustration that the developed optimization method has the capability to search for a better solution provided the objectives and the constraints are realistically defined.

3.4.1 System Definition

The definition of the system for the base model is presented in this section.

3.4.1.1 Reactor Design

MAP1000 is a pressurized water reactor fueled with UO_2 pellets, moderated and cooled with light water. The UO_2 pellets are enriched with U235. Most of the fission interactions in this reactor are initiated by the neutrons having an energy in the thermal range. The following table presents a summary of the design characteristics of the reactor model.

Parameter	**Value**	**Units**
Number of fuel pins per assembly	$152, 176, 192, 220,$	$236, 264$
Number of assemblies	157	
Reactor power (nominal)	3400	MW_{th}
Burn up	23.01	$\frac{MWd}{kg}$
Height of core	42.6	m
Outer radius of core	2.6	m
Enrichment of F1	2.35	atmpct
Enrichment of F2	3.4	atmpct
Enrichment of F3	4.45	atmpct
Density of moderator	0.72	$\frac{gm}{cm^3}$
Density of integral BA	5.42	$\frac{gm}{cm^3}$
Density of discrete BA	7.94	$\frac{gm}{cm^3}$

Table 3.18: MAP1000 design summary.

3.4.1.2 Pincell

At the pincell level, the reactor consists of fuel and water hole pincells. The fuel pincell is composed of UO_2 fuel surrounded by clad. There is a helium gap separating the fuel and the clad. r_f is the radius of the fuel, t_g is the thickness of the gap, and t_c is the thickness of the clad. Some of the fuel rods have an integrated burnable absorber material based coating ($INTBA$). There are a set of burnable absorber pins with discrete burnable poison material with SS304. These are called the discrete burnable absorbers ($DISCBA$). The fuel pincell is shown in Fig: 3.29.

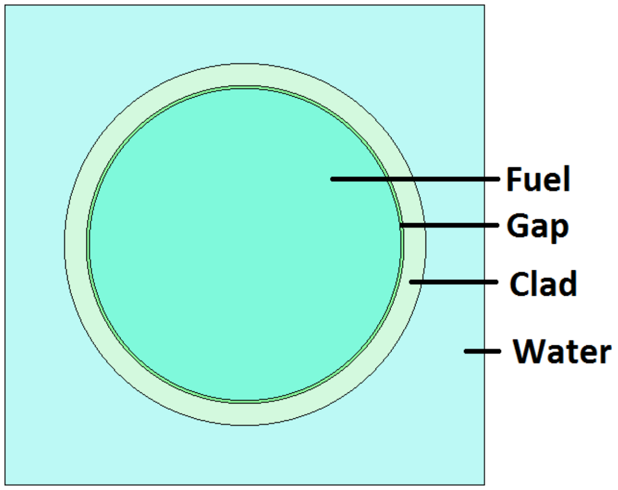


Figure 3.29: MAP1000 fuel pin cell.

The water hole pincell consisted of a water hole surrounded by clad. The water hole is shown in Figs.3.30,3.31. Based on the existing AP1000 model, three types

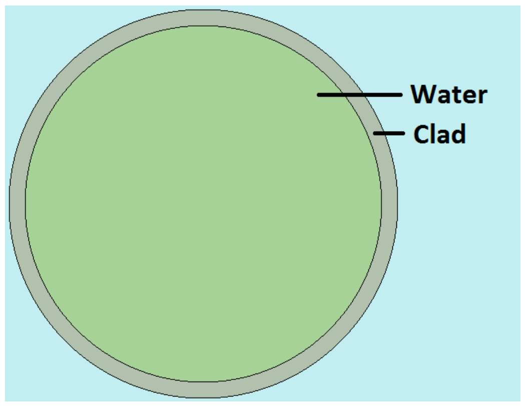


Figure 3.30: MAP1000 water hole pin cell.

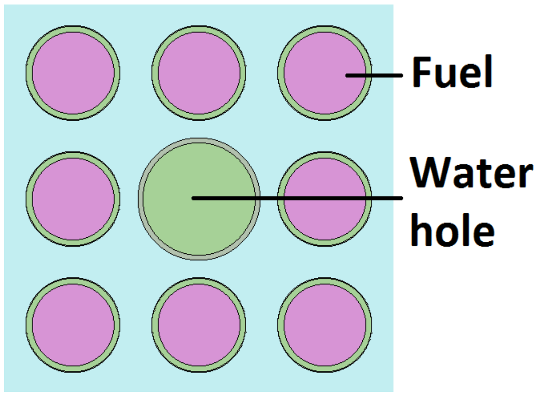


Figure 3.31: MAP1000 fuel and water hole pincell.

of fuel based on the isotopic enrichment of $U235$ have been used in the core. The fuel types have been defined as $F1$, $F2$, and $F3$.

3.4.1.3 Assembly

The PWR assembly consists of a 17×17 lattice of fuel and water hole pin cells. A fuel assembly is presented in Fig: 3.32. The fuel assemblies at the center are surrounded by water assemblies. Water assemblies are similar to fuel assemblies in terms of the assembly dimension and contain only water. Water assemblies serve as a reflector as well as a moderator.

3.4.1.4 Whole Core

The full core is a square lattice comprised of fuel and water assemblies. The reactor model is designed to generate a power P. The radial view of the full core is shown in Fig: 3.33, and the axial view is presented in Fig: 3.34. The burnp versus criticality plot for the base model is shown in Fig: 3.35.

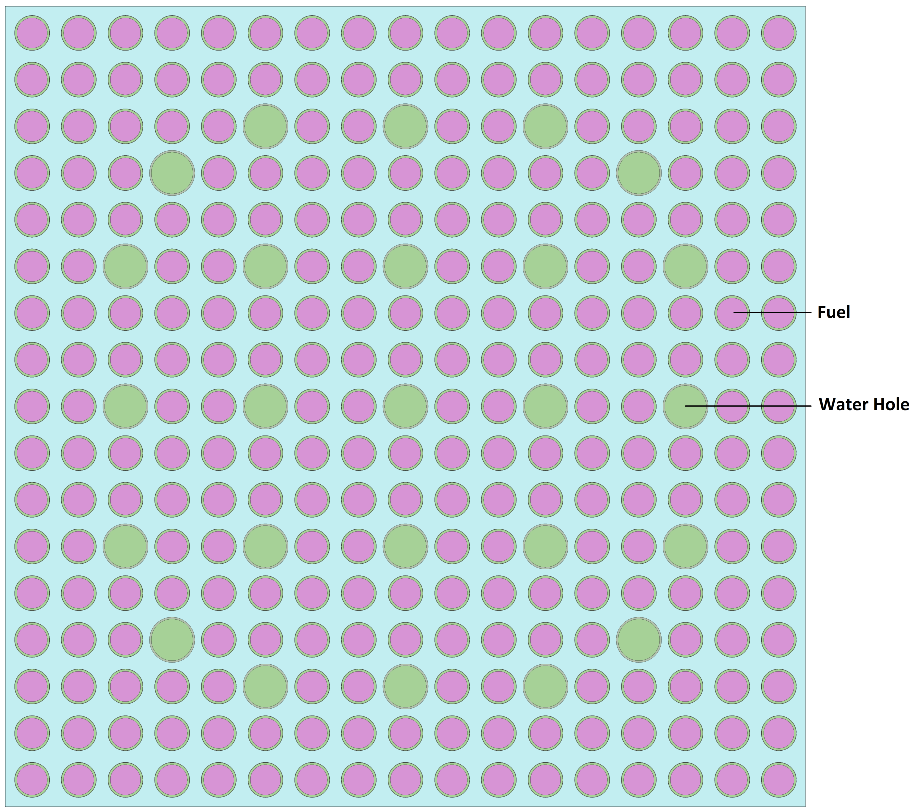


Figure 3.32: MAP1000 fuel assembly.

3.4.2 Optimization Problem Definition

3.4.2.1 Input Design Parameters

A sensitivity study is performed to determine the set of design parameters for the optimization of the core. Based on the sensitivity study, a number of factors have been identified as the input design parameters for the core optimization problem. These parameters are listed in Table: 3.19.

3.4.2.2 Quantities of Interest

For a depletion calculation with B burn steps and a burn step of b ($0 \geq b \leq B$), the QOI for the figure of merit (FOM) are defined as follows,

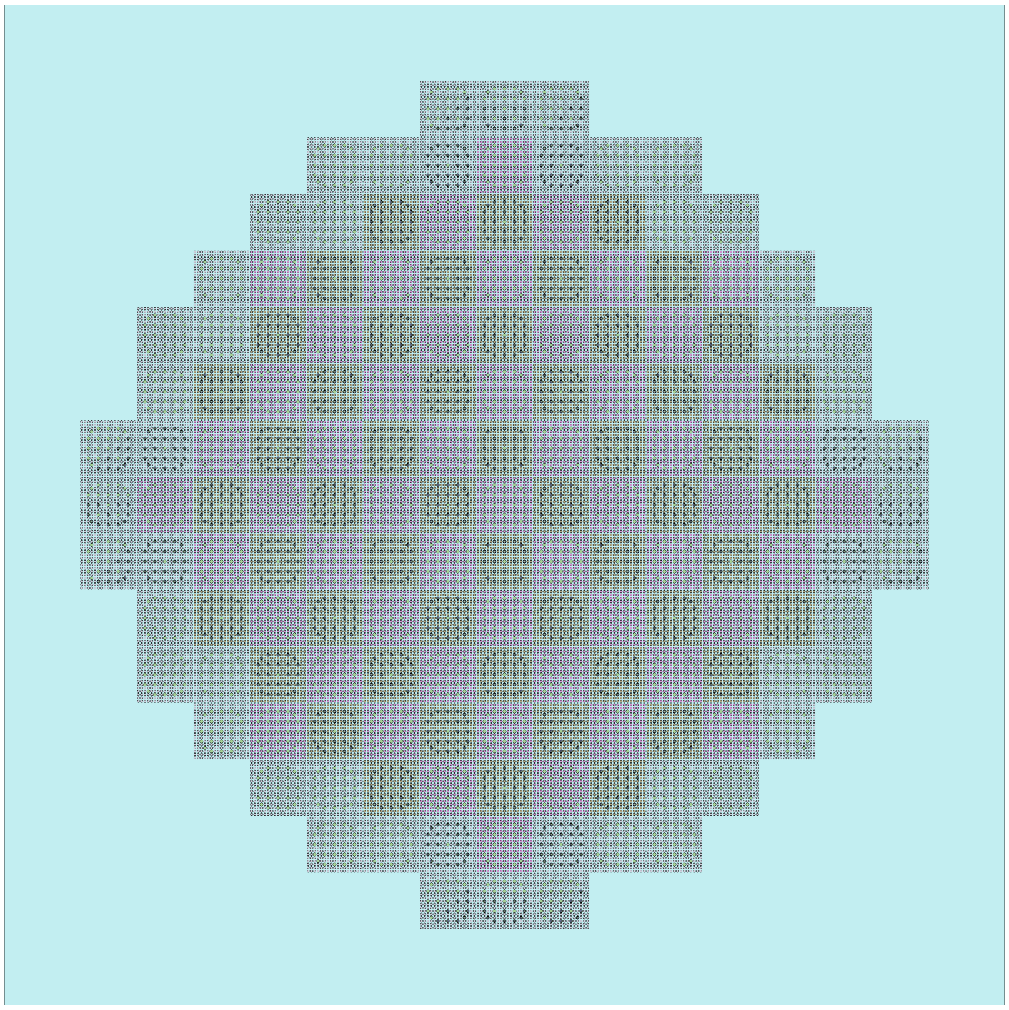


Figure 3.33: MAP1000 full core radial view.

- Criticality at the beginning of life (BOL KEff): $k_{eff,BOL} = k_{eff,0}$
- Radial peaking factor at the beginning of life : $F_{PF,rad,BOL}$
- Burn-up of the reactor model : BU

3.4.2.3 Objectives and Constraint Functions

The following is a list of the objective functions:

1. Minimize peaking factors (radial): The radial peaking factor at the beginning

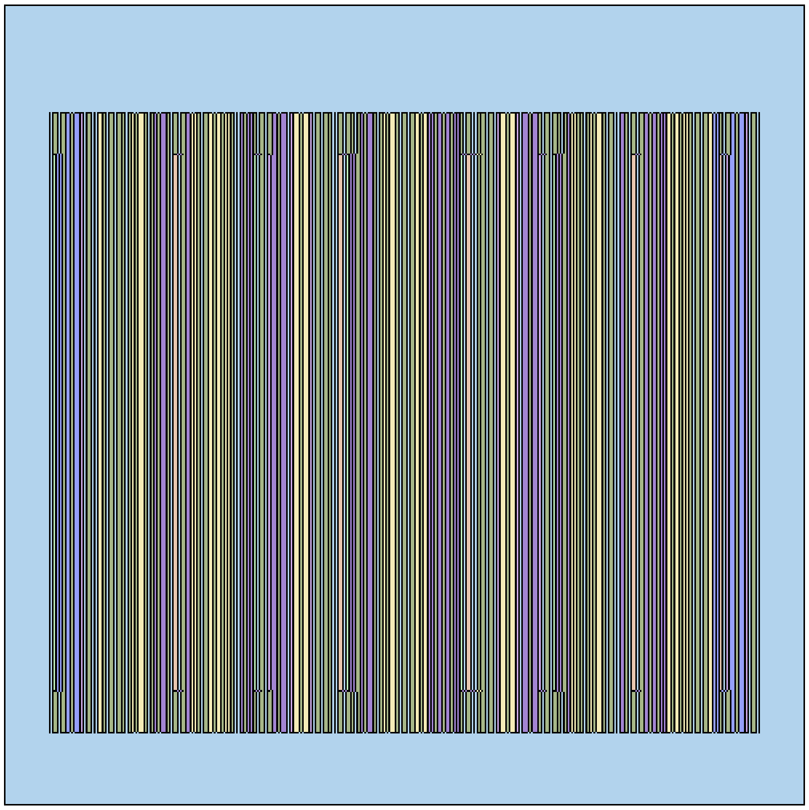


Figure 3.34: MAP1000 full core axial view.

of life is calculated and the objective is to minimize it. The objective function can be defined as:

$$f_1\left(X\right) = \text{minimize}\, F_{PF,rad,BOL}. \tag{3.15}$$

2. Criticality at beginning of life should be close to 1.01: The objective is to have the criticality at the beginning of life of the reactor core close to 1.01:

$$f_2\left(X\right) = k_{eff,BOL} \approx 1.01 \tag{3.16}$$

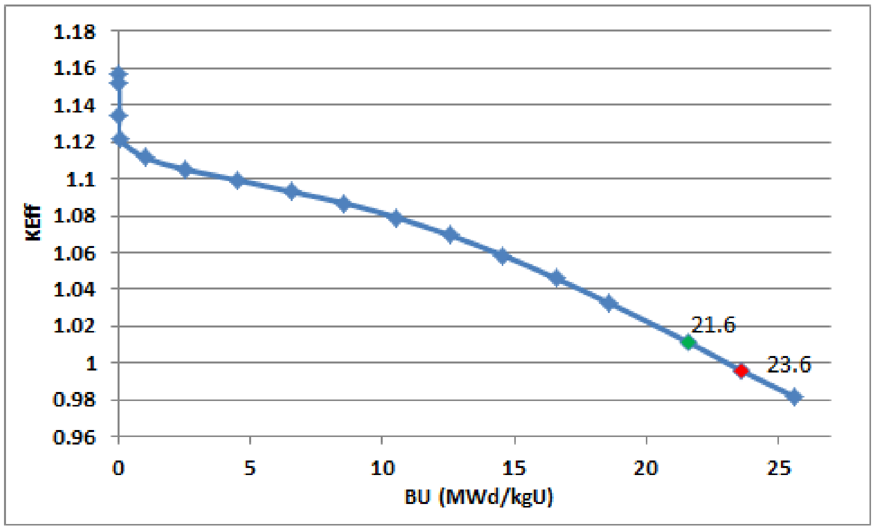


Figure 3.35: BU vs Criticality - MAP1000 base model.

Label	**Description**	**Range**	**Units**
$U235_{F1}$	Isotopic enrichment of F1	$2.15 - 2.55$	atom pct
$U235_{F2}$	Isotopic enrichment of F2	$3.4 - 3.6$	atom pct
$U235_{F3}$	Isotopic enrichment of F3	$4.25 - 4.65$	atom pct
P	Reactor power	$3000 - 3800$	MW

Table 3.19: Optimization design parameters.

3. Maximize burn up: The objective is to maximize the burn up:

$$f_5(X) = \text{maximize}\, BU. \tag{3.17}$$

Label	**QOI**	**Objective**	**Units**
$f_1(X)$	$k_{\text{eff,BOL}}$	$k_{\text{eff,BOL}} = 1.01$	
$f_2(X)$	$F_{\text{PF,rad}}$	Minimize $F_{\text{PF,rad}}$	
$f_3(X)$	BU	Maximize burn-up	

Table 3.20: Optimization objective functions.

Parameter	**Range**
$F(X) = 0$	$k_{\mathrm{eff,BOL}} < 1.0$
$F(X) = 0$	$F_{\mathrm{PF,rad,b}} > 2.5$
$F(X) = 0$	$BU = 0$

Table 3.21: Constraints in PWR optimization

In the ϵ-constraint method, the order of the implementation of the above said objectives determines the importance of that objective with respect to other objectives. E.g. if the order of implementation is $f_1(X), f_1(X), f_3(X)$, then the importance is ordered as $f_2(X) > f_1(X) > f_3(X)$. This importance affects the Pareto optimal solution. In this work, the order of implementation of the objectives is changed and two Pareto optimal solutions are obtained. Table: 3.21 presents the constraints used in the optimization execution. The penalty is imposed by setting the fitness value to be zero.

3.4.3 Iterative Method Parameters

Table: 3.22 presents the values used in the HYBGASA implementation and execution. The parameters include the kernel functions, tuning parameters in GA, ϵ constraint parameters and GPM hyper parameters.

The ϵ constraint method is implemented by using the following constraint limits for the QOIs in the objective functions:

3.4.4 Results

3.4.4.1 Pareto Optimal Solutions and Comparison

Table: 3.24 summarizes the comparison between the base solution and a couple of Pareto optimal solutions. Solution-1 is obtained by performing optimization with the order $F_{PF,rad}$,$k_{Eff,BOL}$,BU. Solution-2 is obtained by performing optimization

	Parameter	**Values**
GA	Number of chromosomes	200
	η	0.05
	ν_{mut}	0.5
	c_k	0.25
	τ_{IF}	1.0E-04
	$\tau_{relDiff}$	1.0E-02
	τ_{NBest}	10
ϵ constraint	intervals	11
	$N_{initial}$	80
Outer	$N_{re-train}$	40
	retrain space	0.5
	convergence threshold	1.0E-06

Table 3.22: Iterative method parameters

Parameter	**Range**
$F_{\mathrm{PF,rad}}$	$1.0 - 2.5$
$k_{\mathrm{eff,BOL}}$	$1.006 - 1.012$
BU	$18.0 - 30.0$

Table 3.23: Constraint limits in the ϵ constraint implementation

with the order BU,$F_{PF,rad}$,$k_{Eff,BOL}$.

Parameter	**Base Design**	**Solution-1**	**Solution-2**	**Units**
En_{F1}	2.35	2.38	2.52	$atmpct$
En_{F2}	3.4	3.51	3.59	$atmpct$
En_{F3}	4.45	4.48	4.60	$atmpct$
P	3400	3138.02	3402.38	MW
k_{Eff}	1.157	1.150	1.161	
$F_{PF,rad}$	1.982	1.829	2.035	
BU	23.01	22.65	23.39	$\frac{MWd}{kg}$

Table 3.24: Comparison of obtained solutions vs base solution.

3.4.4.1.1 Solution-1: The solution has a better peaking factor, and initial criticality as compared to the base solution, but there is a reduction in the burn-up. In terms of the order of the execution of the objective function, the burnup has the least priority. The obtained solution has the enrichments very close to the base model but at a lower power level. This reduction in power can be reflected from a reduction in burnup. The optimization search follows the physics of the behavior of neutrons.

3.4.4.1.2 Solution-2: The solution has a better burnup but compromised on the initial criticality and the peaking factor. In terms of the order of the execution of the objective function, the initial criticality had the least priority. Due to the peaking factor having a higher priority as compared to the criticality, the peaking factors between the solution and the base solution are pretty close to each other. There were a couple of other solutions with a higher burnup but those were rejected because of their proximity to criticality and the peaking factor. It can be observed that the enrichments are very close to the upper bound of the design parameters, this means that the criticality and the peaking factors get affected from a physics point of view.

3.4.4.2 Sensitivity Studies and Parameter Search

The first step involves an extensive search for the values of the parameters used in the regression model and the ϵ-constraint method. Search studies have been done to determine the following parameters/functions:

1. Kernel function for GPM
2. GPM hyper-parameters
3. ϵ intervals

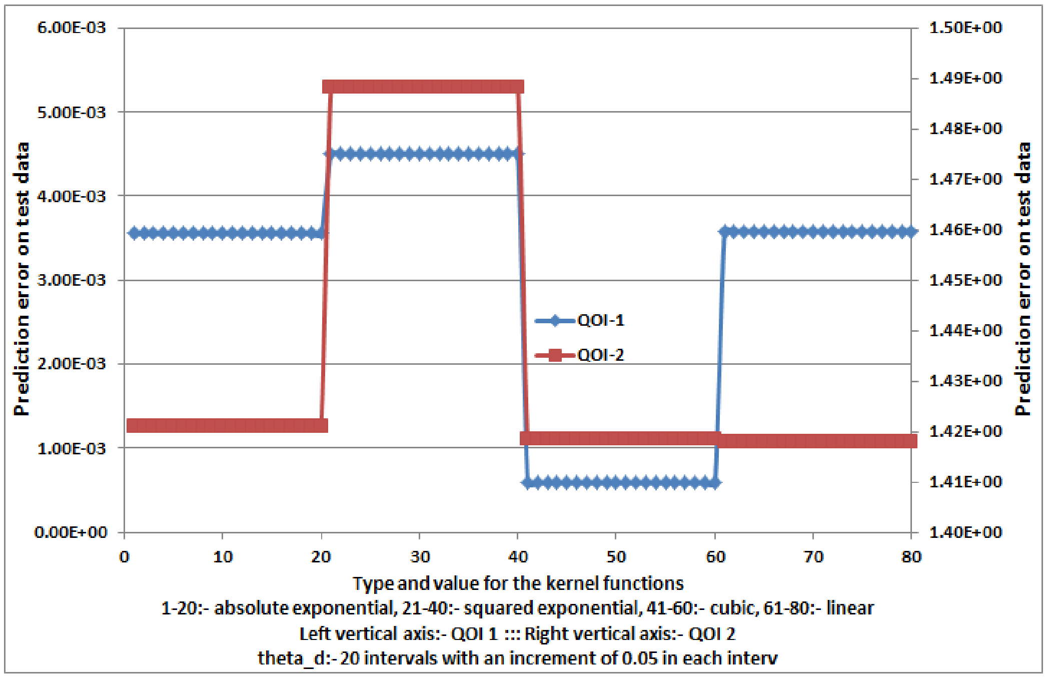
3.4.4.3 GPM Kernel Function and Hyper Parameter Search

A grid based search is performed to determine the kernel function and the hyper parameter that performs the best prediction for a QOI in every outer iteration. The grid search is given in Table: 3.25: Based on the combinations of the above

Kernel Function	θ_d **grid**
Absolute exponential	0.05-1.0 (Intervals-0.05)
Squared exponential	0.05-1.0 (Intervals-0.05)
Cubic	0.05-1.0 (Intervals-0.05)
Linear	0.05-1.0 (Intervals-0.05)

Table 3.25: Kernel function and hyper-parameter search grid for the MAP1000 problem

grid, a search is performed and the results of the search at the end of the first outer iteration is given in Fig: 3.36: The first 20 points on the x-axis represent the "absolute exponential", next 20 points represent the "squared exponential", the following 20 represent "cubic" and the last 20 represent the "linear" kernel function. For each function, a set of twenty values for θ_d is analyzed, and the MPE for each of these are shown on the y-axis. θ_d ranges from 0.05 to 1.0 with an interval of 0.05. The colored lines represent a the QOIs. It is evident that different QOIs have their own choice of the kernel function that gives the best predictions. It is assumed that the MPE in the predictions from the GPMEM built using a function and θ_d is directly proportional to the quality of the predictions i.e. a function-θ_d pair that produces the least MPE is the best prediction model. For QOI-1, "squared exponential" kernel performs the best as compared to other kernel functions, while for QOI-3, "cubic" has the best predictions. The next step is to determine the exact value of θ_d that gives the best



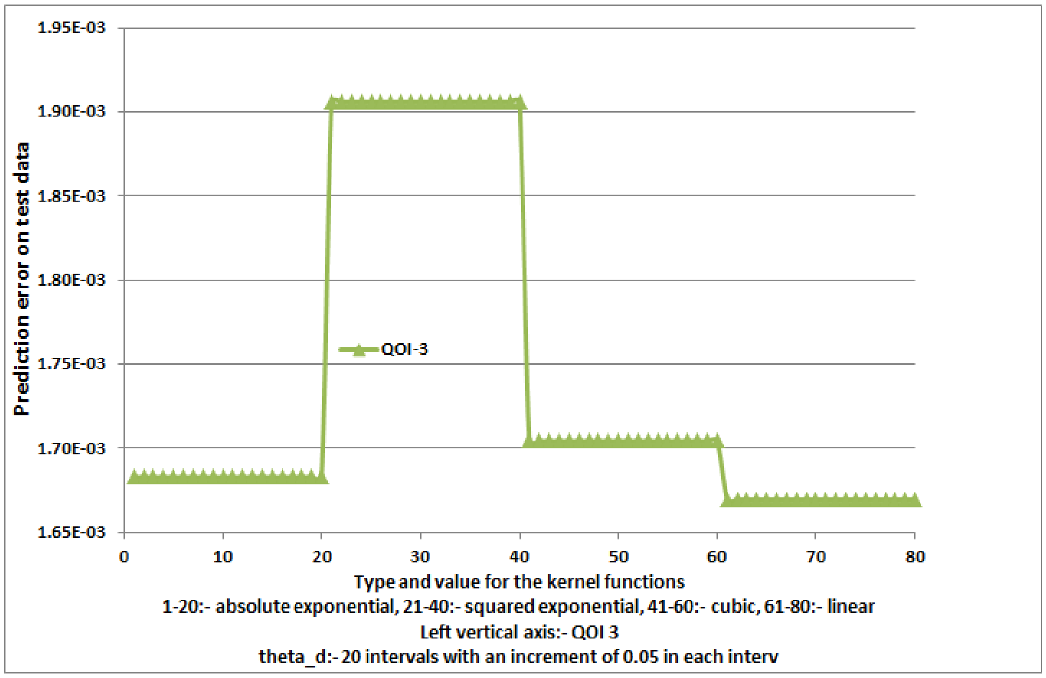


Figure 3.36: Test prediction error versus kernel function and hyper parameter for the QOIs at the beginning of the first outer iteration Top: QOI-1,2 Bottom: QOI-3

predictions for a QOI. The choice of θ_d along with the best kernel function for four QOIs in outer iteration-1 is given in Fig: 3.37 and Table: 3.26. Each of the sub-plots

Outer Iteration	**QOI**	**Kernel Function**	θ_d
1	1	Cubic	0.85
	2	Linear	0.35
	3	Linear	0.5

Table 3.26: Kernel function and hyper-parameter results

in Fig: 3.37 represent the best performing kernel function for each of the QOIs. The color of the subplots are consistent with the color of that specific QOI in Fig: 3.36. It should also be noted that the kernel functions and the hyper parameter depends on the training data, and therefore the choice is made at the beginning of every outer iteration. From the analysis of other outer iterations it is observed that the choice of the kernel function that gives the best prediction for a specific QOI does not depend on the outer iteration. Rather, the choice of θ_d depends on the outer iterations. This can be inferred from the fact that the regression surface is significantly affected because of re-training the model. The results for other outer iterations are shown in the Appendix.

3.4.4.4 ϵ Intervals

A sensitivity study is performed by changing the number of ϵ intervals and investigating the change in the fitness values. The sensitivity study is performed during the beginning of every outer iteration when the training sample size is updated. A threshold value of $1.0E-06$ is used to determine the number of intervals i.e. the number of intervals corresponding to a relative change in the fitness value that is less than or equal to the threshold is the final number of intervals in that outer iteration.

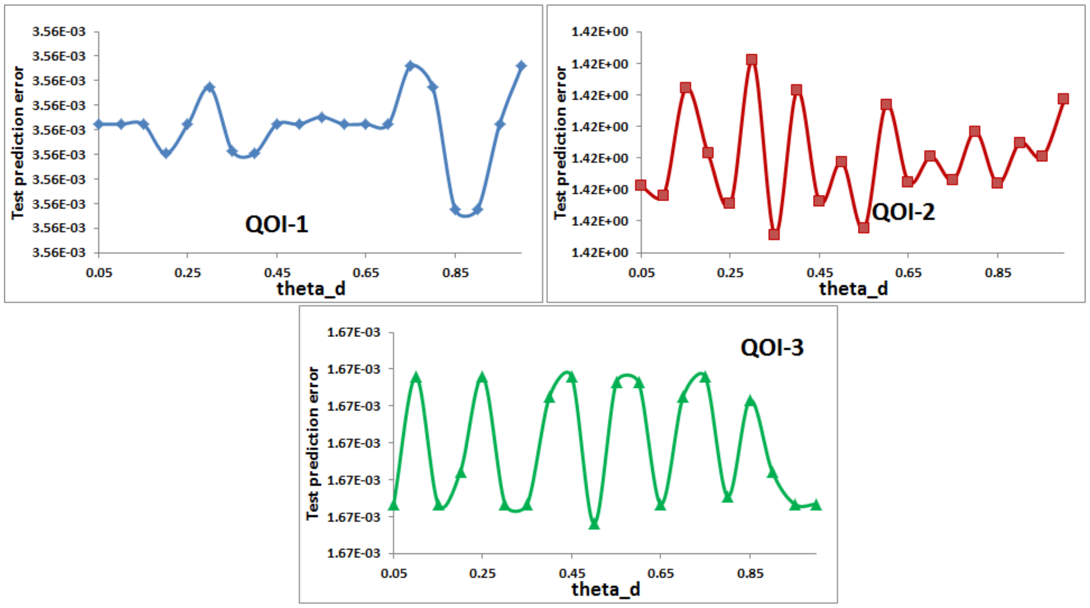


Figure 3.37: Kernel function and hyper parameter versus test prediction error for a set of QOIs at the beginning of first outer iteration. The θ_d plots correspond to the best performing kernel function for each of the QOIs.

From Fig: 3.38 the number of intervals is chosen as 5. It is not known whether the

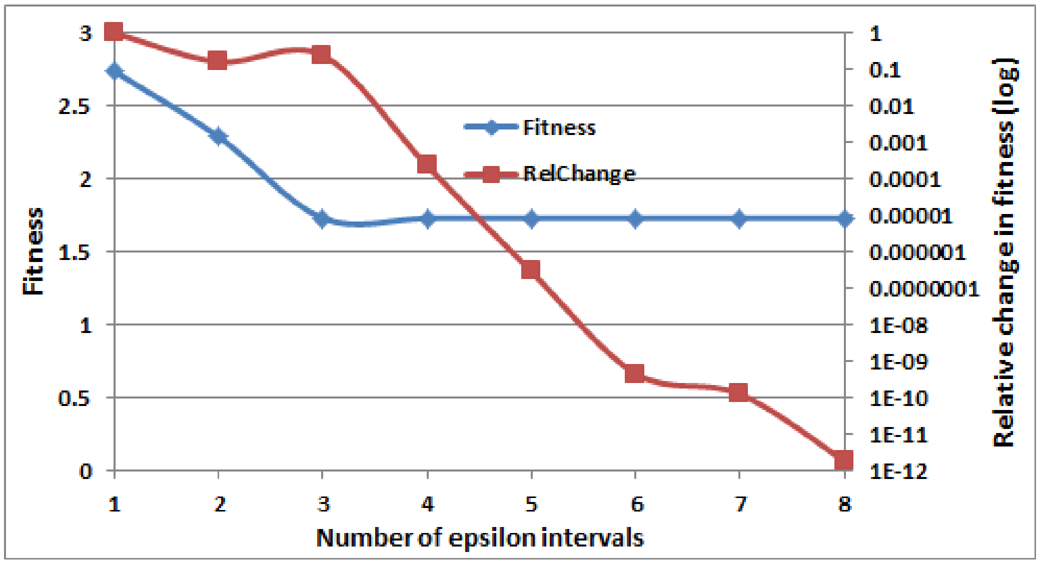


Figure 3.38: ϵ intervals versus fitness - Outer iteration - 1

defined system has a convex objective space or not, therefore the ϵ sensitivity study helps to investigate if there is concavity in the space. A convergence of the fitness value to a saturated point with the change in ϵ intervals indicate that the concavities have been taken care of in the intervals. With the number of intervals at 5, the iterations for each of the objectives in defined as:

- Objective-1: GA is performed to optimize objective:1 with other objectives as constraints. There are a total of 25 epsilon intervals in which the single objectives optimization is performed.
- Objective-2: GA is performed to optimize objective:2 with other objectives as constraints. There are a total of 5 epsilon intervals in which the single objectives optimization is performed.
- Objective-3: GA is performed to optimize objective:3.

3.4.4.5 Genetic Algorithms

Figs: 3.39 - 3.40 show the convergence of the first four QOIs during the last outer with a single epsilon interval. The QOIs are BU, $k_{eff,EOL}$, and F_{PF}. This is for solution-1 where the priority is the peaking factor. The best solution out of all the chromosomes in a particular generation is saved and its value across several generations is presented. The stopping criteria and search direction towards GA convergence in the last outer iteration is presented. The convergence of the mean and the best solution of all the chromosomes in each generation is used in the determining the right time to stop GA. Upon convergence of the GA, the solution of all the chromosomes move towards the optimal solution, and therefore the mean traverses towards a saturated value. One of the stopping criteria is to track the generations when the relative difference goes below 0.01. The improvement in fitness and the

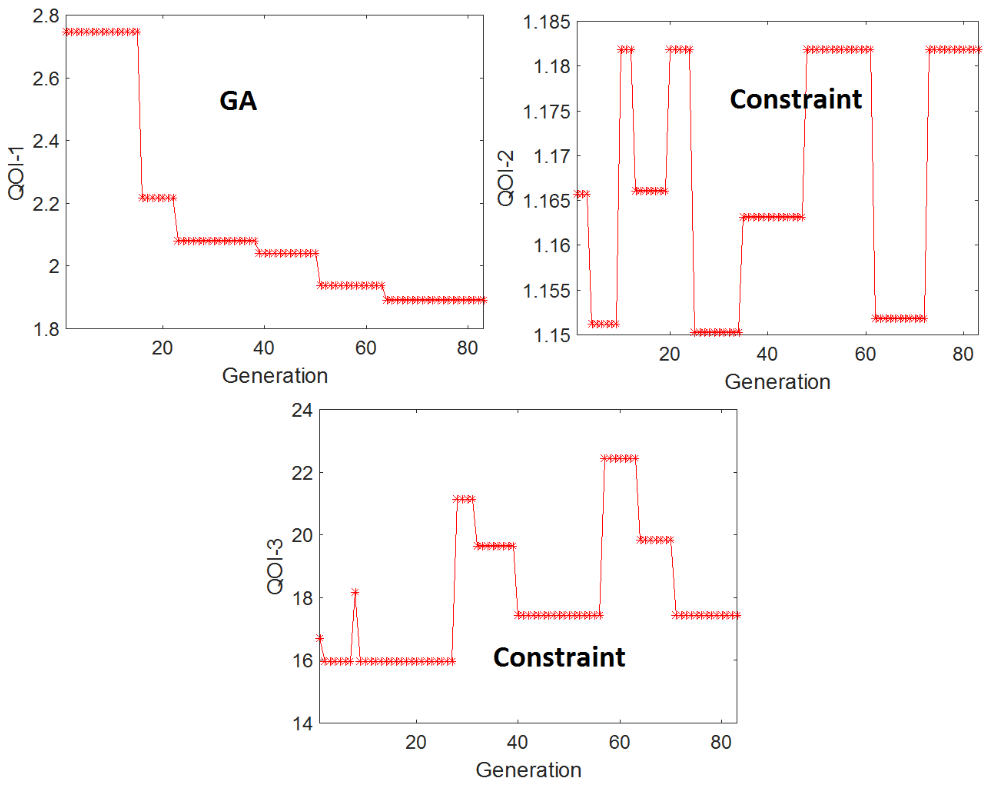


Figure 3.39: Convergence of GA for three QOIs in the last outer iteration for a single epsilon interval, Objective:1

direction of the best fitness is considered to be a stopping criteria along with the relative difference criteria. There is not a significant relationship between an outer iteration and the number of GA generation in that outer iteration.

3.4.4.6 Outer Iterations

The IHGOM method for Solution:1 converged after 8 iterations. With an initial sample set of 80, and re-sampling with 40 samples, eight outer iterations involved an execution of a total of 360 samples. Table: 3.27 shows the Pareto optimal values for the design parameters at the end of the convergence of the outer iterations for Solution:1. Table: 3.28 shows the converged values for the QOIs after the end of all

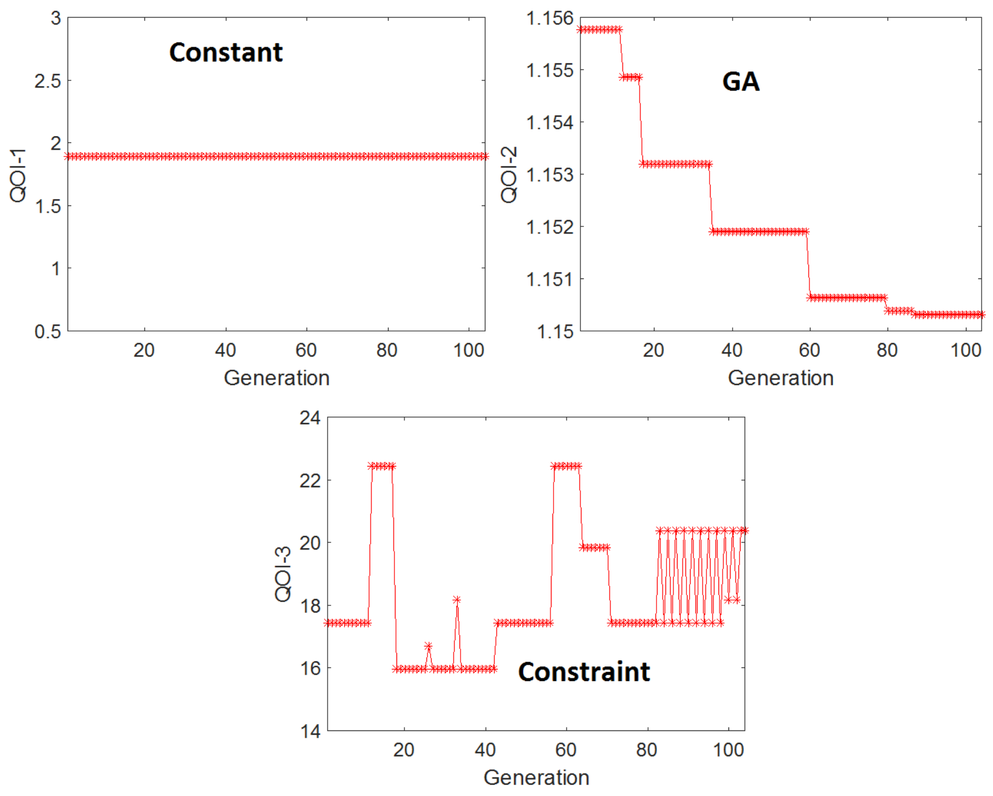


Figure 3.40: Convergence of GA for three QOIs in the last outer iteration for a single epsilon interval, Objective:2

Parameter	**Values**
$U235_{F1}$	$3.3807 + 00$
$U235_{F2}$	$3.5123E + 00$
$U235_{F3}$	$4.4843E + 00$
P	$3.1380 + 03$

Table 3.27: A Pareto optimal solution after the convergence of the outer iterations, Solution:1

the outer iterations. This table contains the actual values and the relative error in predictions obtained from the execution of the actual system.

The IHGOM method for Solution:2 converged after 10 iterations. With an initial

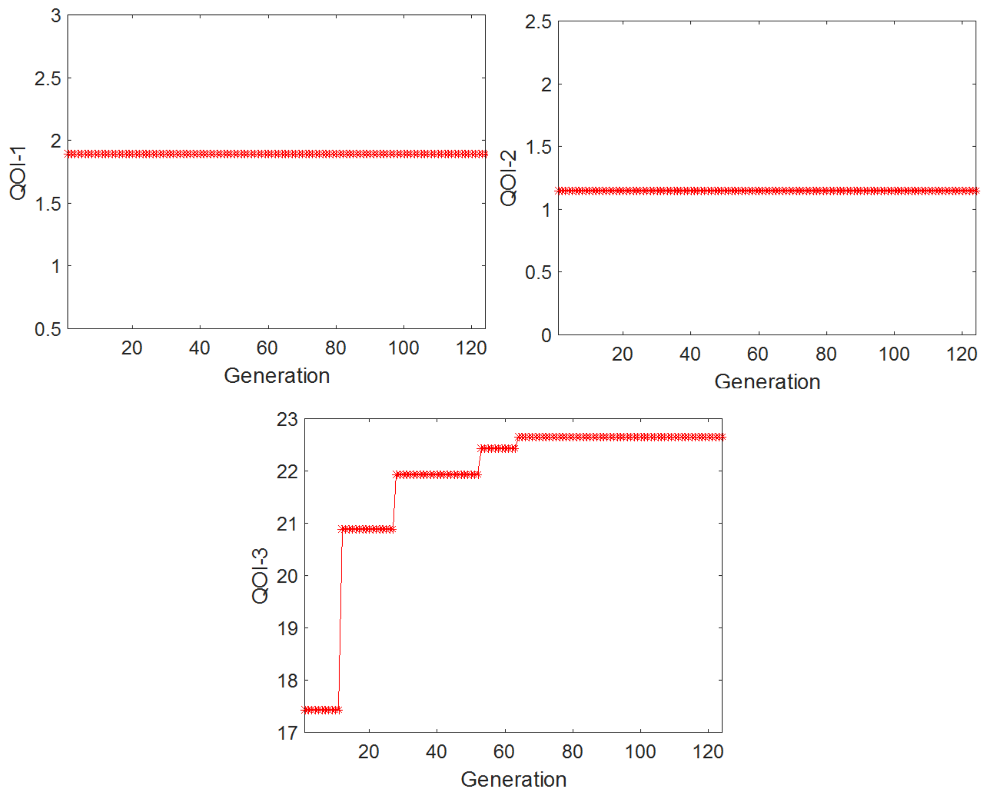


Figure 3.41: Convergence of GA for three QOIs in the last outer iteration for a single epsilon interval,Objective:3

Parameter	**QOI Predictions**	**QOI Actual values**	$\epsilon_{\text{predErr}}$
$F_{PF,rad}$	$1.8923E+00$	$1.8924E+00$	$2.1137E-05$
$k_{eff,BOL}$	$1.1511E+00$	$1.1503E+00$	$7.1285E-04$
BU	$2.2637E+01$	$2.2654E+01$	$5.3819E-04$

Table 3.28: A Pareto optimal solution - QOIs after the convergence of the outer iterations - Solution:1

sample set of 80, and re-sampling with 40 samples, eight outer iterations involved an execution of a total of 440 samples. Table: 3.29 shows the Pareto optimal values for the design parameters at the end of the convergence of the outer iterations for

Solution:2. Table: 3.30 shows the converged values for the QOIs after the end of all

Parameter	**Values**
$U235_{F1}$	$2.5208 + 00$
$U235_{F2}$	$3.5987E + 00$
$U235_{F3}$	$4.6024E + 00$
P	$3.4023 + 03$

Table 3.29: A Pareto optimal solution after the convergence of the outer iterations, Solution:2

the outer iterations. This table contains the actual values and the relative error in predictions obtained from the execution of the actual system.

Parameter	**QOI Predictions**	**QOI Actual values**	$\epsilon_{\text{predErr}}$
$F_{PF,rad}$	$2.0358E + 00$	$2.0366E + 00$	$4.1261E - 04$
$k_{eff,BOL}$	$1.1615E + 00$	$1.1614E + 00$	$6.0267E - 05$
BU	$2.3387E + 01$	$2.3375E + 01$	$4.9601E - 04$

Table 3.30: A Pareto optimal solution - QOIs after the convergence of the outer iterations - Solution:2

Fig: 3.42 show the convergence of the outer iteration for QOI-1. The table and the figure compare the test MPE reduction between local re-sampling and global re-sampling. This shows that local re-sampling is more effective as compared to global re-sampling in re-training the GPM for predictions. Fig: 3.43 shows the re-sampling space of the design parameters versus outer iteration. With the outer iteration, the difference between the lower and the upper bounds for all the design parameters decreases.

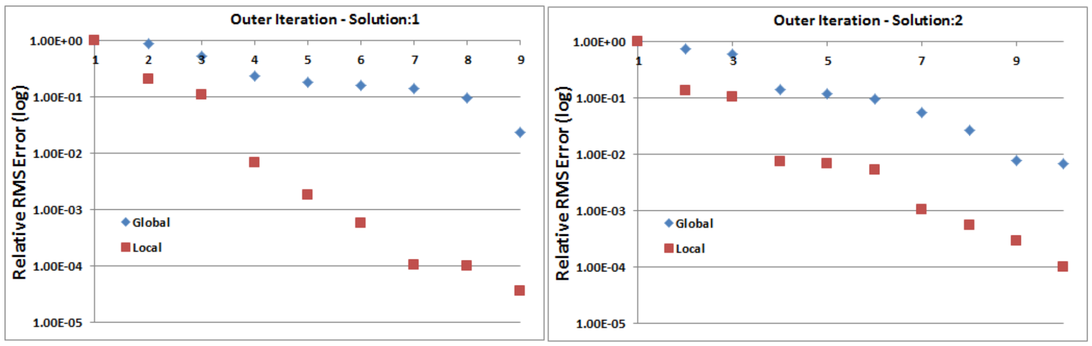


Figure 3.42: Test predictions error in local re-sampling and global re-sampling versus outer iteration for QOI-1

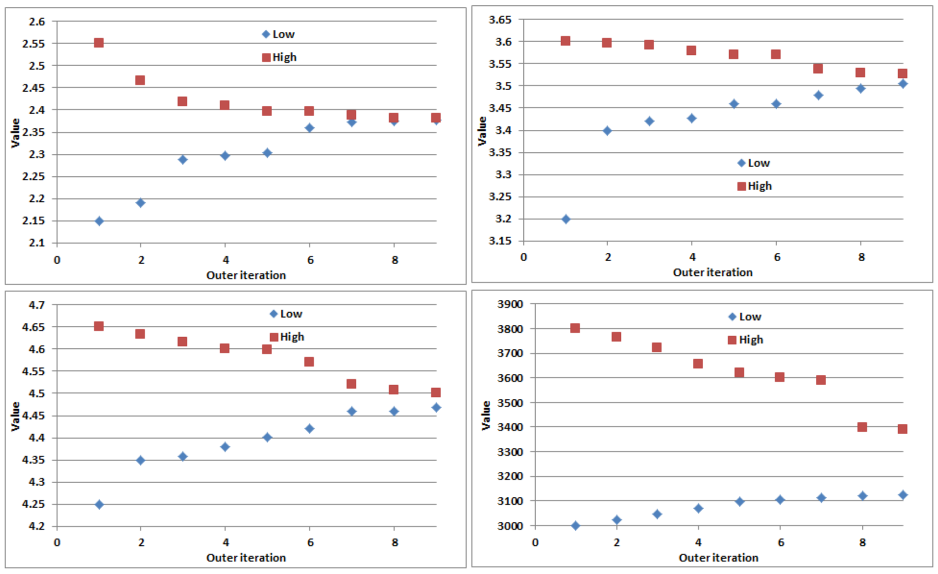


Figure 3.43: The design parameter bounds versus outer iteration

3.4.4.7 Discussion

The novelty of the method is described in the following discussion,

3.4.4.7.1 Speed: The comparison between local re-sampling and global re-sampling (Fig: 3.42) shows that the method which used local re-sampling reduces the MPE at a significant speed as compared to the traditional global re-sampling process.

3.4.4.7.2 Accuracy: The accuracy of the method is demonstrated based on the accuracy of the predictions at the end of the outer iterations. The accuracy is measured in terms of the prediction error i.e. how far the predictions for the QOIs are from the values obtained from the experiment/simulation. This is shown in Table: 3.28, 3.30.

3.4.4.7.3 Search Efficiency: The optimization capability of the developed method is shown by comparing the solution obtained for the QOIs (Table: 3.28, 3.30) and the optimization objectives and the constraints, as presented in Table: 3.20 and Table: 3.21. Comparing the results to the base solution we see that the order of execution og the objectives affect the Pareto optimal solution. All of the objectives and the constraints for the problem are satisfied and the values for the design parameters are within the limits. The discussion follows the same argument presented in the GCFBR problem. The optimal solution for the problem is not known, and therefore it is not possible to say that the optimal solution has been determined. However, a qualitative approach towards the optimality of the solution show that a good solution based on the defined objectives and the constraints are obtained. The search efficiency has been measured in terms of how efficient the obtained optimal solution is based on the objectives and the constraints. In the above comparison, a discussion on the proximity of the solution QOIs to the objectives is presented. The sustainability objective is met by the determination of a higher value for the burn-up. The safety objectives are met by the determination of a lower value for the peaking factors.

4. CONCLUSIONS AND PROSPECTIVE APPLICATIONS

This chapter presents the summary, conclusions, future work and prospective applications of the method developed in this work.

4.1 Summary and Conclusions

A novel optimization method (IHGOM) has been developed and its applications in reactor design has been demonstrated. IHGOM is a black-box method that is applicable to any complex system. The method is implemented in a modular way such that any optimization problem of a complex system is solved by defining a set of design parameters, QOIs, objectives and constraint functions. Ackley's function and an asymmetrical 2-D test problem are used to test the search capability of IHGOM. Two demonstration systems have been characterized and the application of IHGOM in solving optimization problems defined on those systems has been demonstrated. The demonstration problems with several assumptions and simplifications, are a representative form of the actual physical problem and have been developed to show the effectiveness of the iterative optimization method. The objective functions for the demonstration problems have been developed such that they have a mixture of maximization and minimization individual objectives with the task to search for a Pareto optimal solution. The shape of the objective space, whether it is convex or concave is not known, therefore ϵ constraint method has been implemented instead of the standard weighted approach in GA.

The effectiveness of the IHGOM in terms of speed, accuracy, and search efficiency has been presented. In fact, the optimal solution for the demonstration problems are not known, therefore a qualitative approach by analyzing the proximity of the values for the QOIs to the objectives and the constraints is used to understand the search efficiency and optimality of the solutions. The method has been successful in the determination of a couple of Pareto optimal solutions starting from a base model of an AP1000 reactor design. Based on the order of execution of the objectives i.e. the priority of the objectives, the Pareto optimal solutions were generated and their proximity for the base solution has been analyzed. The method has been successful in solving the Ackley's test function and the asymmetrical 2-D test function in determining the global optima in a space with having several local optima regions, which is an indication that the search functionality is effective.

This work lacks an extensive sensitivity study and parameter search which opens doors for future work. The definition of design parameters, their bounds, objectives and the constraints play an important role in the search. Particularly, due to the fact that the optimizer has no information about the underlying physics, the above parameters direct the search and define how the design parameter bounds are affected. In this iterative approach the design parameter bounds play a significant role in determining the speed of the search. Figures and analysis on the differences in local resampling versus global resampling provide sufficient evidence that the changes in bounds affect the speed in the search. The developed method shows a novel approach towards optimization and has the potential to be implemented in several other domains.

4.2 Future Work

The current work has opened new avenues in the development of a fast and accurate method for optimization of complex systems. Future work include a set of suggestions that can improve the efficiency of the developed method.

4.2.1 Sensitivity Studies of GA Parameters

The HYBGASA model has several tuning parameters that affect the accuracy of the optimization search in terms of search efficiency and the convergence speed. In particular, analysis of operators like cross over and mutation, selection of the number of chromosomes, and a robust stopping criteria for convergence are a subset of tuning parameters in GA. The type of selection method, crossover and mutation in GA plays a significant role in the search efficiency. A sensitivity study to analyze the effect of these operators is important in choosing the same for the demonstration problems.

4.2.2 Adaptive and Smart Parameter Search for GPM

The kernel function is important in the prediction ability of the regression model based on the Gaussian Processes. The choice of the kernel function and the hyper parameters in the kernel function play a significant role in the regression models. Adaptive methods have been used to search for the optimum hyper parameter for every regression model. Similarly, learning based methods can be implemented to adaptively update the values for the hyper parameters based on the size of training data and the size of the feature set. Research has been done in the development of customized kernel functions [40, 41], but the customization is usually based on the underlying physics. Therefore, a generalized approach is very difficult to build.

The GPM Kernel function sensitivity search can provide information about the behavior of the dependent variable. E.g. if the kernel function chosen adaptively is a an exponential kernel, then the underlying function can be thought of a s a smooth function. This information can be re-used in reducing the complexity of the regression model leading to an increase in speed.

4.2.3 Derivatives in the Regression Model

Implementation of the local derivatives of the observation in the learning model increases the prediction accuracy of the regression model. The implementation follows the underlying idea that the derivative of a Gaussian process is a Gaussian process. With reference to [42], the covariances between function values and derivatives and between derivatives are given by:

$$K = \begin{vmatrix} K_{ff} & K_{fD} \\ K_{Df} & K_{DD} \end{vmatrix}, \tag{4.1}$$

where K_{ff} is the covariance between the parameters, K_{fd} is the covariance between parameters and derivatives, and K_{dd} is the covariance between derivatives given by:

$$K_{ff}^{ij} = Cov\left(f_i, f_j\right) = k\left(x_i, x_j\right) \tag{4.2a}$$

$$K_{fD}^{ijd} = Cov\left(f_i, \frac{\partial f_j}{\partial x_d}\right) = \frac{\partial k\left(x_i, x_j\right)}{\partial x_d} \tag{4.2b}$$

$$K_{DD}^{ijde} = Cov\left(\frac{\partial f_j}{\partial x_d}, \frac{\partial f_j}{\partial x_e}\right) = \frac{\partial^2 k\left(x_i, x_j\right)}{\partial x_d \partial x_e}. \tag{4.2c}$$

The joint covariance matrix for the functions and the derivatives in an expanded form are given by:

$$\hat{K} = \begin{bmatrix} K_{ff} & \begin{bmatrix} K_{fD}^{1} & K_{fD}^{2} & .. & K_{fD}^{K} \end{bmatrix} \\ \begin{bmatrix} K_{fD}^{1} \\ K_{fD}^{1} \\ .. \\ K_{fD}^{K} \end{bmatrix} & \begin{bmatrix} K_{DD}^{1,1} & K_{DD}^{1,2} & .. & K_{DD}^{1,K} \\ K_{DD}^{2,1} & K_{DD}^{2,2} & .. & K_{DD}^{2,K} \\ .. & .. & .. & .. \\ K_{DD}^{K,1} & K_{DD}^{K,2} & .. & K_{DD}^{K,K} \end{bmatrix} \end{bmatrix} \tag{4.3}$$

where K is the dimension of the input parameters. The output parameters vector is updated to add the dervatives of the observations as,

$$\hat{F} = \left[F \quad \frac{\partial f(X)}{\partial x_1} \quad \frac{\partial f(X)}{\partial x_2} \ldots\ldots \frac{\partial f(X)}{\partial x_L} \right]^T, \tag{4.4}$$

where $\frac{\partial f(X)}{\partial x_1}$ is the local derivative obtained using acceleration methods, if y_i is an output parameter represented in a different form than $F(X)$. The values for these derivatives of the observations are obtained from standard acceleration methods that build local derivatives. A comparison between several local derivative methods and their effects on the prediction ability of GPMs is a significant contribution to the research community.

4.2.4 Sensitivity Studies of the Parameters in the Outer Iteration

Initial sample size, re-sample size and re-sampling rate play a significant role in the speed, convergence and the accuracy of the outer iterations. These parameters affect the fitness of the regression model. In the modeling of the surrogate models there is always a fear of the regression model being under-fit or over-fit. Under-fit or over-fit models tend to increase the error in the predictions. Therefore, for the most accurate predictions, the regression model needs to be a “good” fit. A sensitivity study to analyze the effect of these parameters on the fitness of the regression model is necessary for a good predictive model.

4.2.5 Regression Methods

Comparison of the performance of several other regression methods such as Ridge, Lasso, Markhov Chain Monte Carlo (MCMC), logistic, random forests, decision trees etc.. as compared to GPM is important in the development of a more accurate, robust and faster predictive model. The regression methods are dependent on the feature size, data size and the underlying physics of the data. A rigorous approach in choosing the regression method and the corresponding hyper-parameters will affect the scalability of the developed method.

4.2.6 Uncertainty Propagation in Heuristics Based Optimization Methods

Real-time physical applications have uncertainties in the experimental data. Any optimization problem based on the experimental data is affected by the uncertainties in the search for the optimum design space. The method of the propagation and the uncertainties across generations in heuristics methods such as GA has not been explored yet.

4.3 Prospective Applications

The iterative optimization method developed in this research has prospective applications in a number of research areas. A few of the prospective application areas identified during the course of the work are described in this section.

4.3.1 Fuel Loading and Shuffling Optimization

The optimization method is applicable in fuel selection, loading position and shuffling strategy in nuclear reactors. Selection of fuel, particularly used fuel from storage casks and/or fresh fuel elements involves a multi parameter search based on multiple objectives. Researchers have used GA and SA in fuel loading and shuffling optimization in PWRs [43, 44, 6, 6, 45]. A coupled approach presented in this dissertation is yet to be used by the optimization research community. A presented in this work, the coupled approach has the capability to search for a Pareto optimal solution at a significant speed.

4.3.2 Position of Control Rods and Other Burnable Absorbers

The position of the control rods (CR) in the design of nuclear reactors depend on the local flux, excess reactivity, peaking factors, desired core lifetime, etc.. Therefore the location of CRs and the amount of burnable absorbers is an optimization design problem. The objective space is dependent on the desired burn up, power, safety margin, and fuel inventory.

4.3.3 Fuel Cycle Analysis and Fuel Economy

The relation between fuel economy and reactor power is a multi-design-multi-objective optimization problem. Several machine learning based optimization methods have been explored in the fuel cycle optimization of AP1000 reactors [46, 47] by Westinghouse. However, there is a need in the design and modeling of a robust mult-objective optimization method that couples physics based modules such as neutronics and thermal-hydraulics with fuel cycle and economics.

4.3.4 Position of Detectors for Spectrum Reconstruction

Machine learning methods are applicable in determining the positioning of the detectors in the reconstruction of flux for safety and operation purposes. During the operation of a nuclear reactor, there is a shift in the peaking factors based on the control elements (control rods and burnable absorbers) and fuel burnup. This is a predictive behavior and a regression model that can be fit to emulate the behavior. A basic demonstration has been presented in this dissertation. There is a need for a robust model development with several design parameters and QOIs applicable to the real physical problems. This predictive model can be used to determine the ideal locations for detectors for spectrum reconstruction.

4.3.5 Prediction and Forecasting of Experiments in Test Reactors

Irradiation experiments in test reactors are expensive and a predictive forecasting model helps in cost-savings for experiments. A learning based experiment design and cycle length predictions help in the implementation of efficient experiments that solve the purpose of the irradiation and save fuel. A prospective application has been identified in the fuel selection and loading patterns for the Idaho National Laboratory (INL) Advanced Test Reactor (ATR) [48] . The ATR at the INL is a pressurized, light-water moderated and cooled, beryllium-reflected, highly enriched uranium fueled reactor with a maximum operating power of 250 MWth. The following Core Physics Analysis (CPA) heuristic outputs would be used to evaluate a QOI for a randomly sampled set of fuel element loadings:

- Estimated critical position
- Estimated lobe-power split
- Estimated radial and axial power peaking factors
- Estimated fission density at the end-of-cycle (EOC)

A subset of design parameters are used as inputs to the above heuristics:

- Predicted cycle length
- Reactivity worth for each capsule
- Number and position of recycle fuel elements to be identified and used from the CANAL
- Number and position of the fresh fuel elements in the reactor core

The above inputs dictate the U235 and B10 loading of each fuel element and by extension each lobe. The estimated heuristic outputs will be based on a surrogate or empirical model of the ATR. This surrogate model is built as a standard database, prior to implementation of the Total Heuristic Evaluator Critical Process Transformer (THECPT code) to increase the ATR fuel efficiency.

4.3.6 Prediction of the Source of Radioactive Material in Nuclear Forensics

The possibility of weapons-grade Plutonium in foreign nuclear fuel cycle reactors is a global security concern. One of the challenges associated with this is the identification of the source should the illicit transfer of weapons-grade plutonium be interdicted. The source includes the originating country, fuel cycle reactor and irradiation history. Predictions of the source is a multi-parameter regression problem. A logical set of tasks involves:

1. Design and development of a set of experiments and simulations to develop the training data set.
2. Build a regression based predictive model based on the training data set.
3. Isotope separation and identification of radioactive nuclides from the interdicted material.
4. Predict the source of the interdicted material from the predictive model developed in Step:2.

4.3.7 Composite Fuel Optimization

In the development of proliferation resistant and sustainable fuels, "composite fuel" [49, 50] have played an important role. Composite fuels (CF) consist of matrix and multiple seed materials, for example metallic U alloys and mixed oxide (MOX) mixed at microscopic level. The objectives are to increase fuel resource utilization by extending fuel burnup while simultaneously satisfying core safety requirements and non-proliferation standards. The sizes of the microscopic elements along with the composition pose a muli-objective optimization problem.

REFERENCES

[1] Cláudio MNA Pereira and Celso MF Lapa. Coarse-grained parallel genetic algorithm applied to a nuclear reactor core design optimization problem. *Annals of Nuclear Energy*, 30(5):555–565, 2003.

[2] Cláudio Márcio do Nascimento Abreu Pereira, Roberto Schirru, and Aquilino Senra Martinez. Basic investigations related to genetic algorithms in core designs. *Annals of Nuclear Energy*, 26(3):173–193, 1999.

[3] Brian Vincent Haibach and Madeline A Feltus. A study on the optimization of integral fuel burnable absorbers using the genetic algorithm based cigaro fuel management system. *Annals of Nuclear Energy*, 24(6):439–448, 1997.

[4] M Cantoni, Marzio Marseguerra, and Enrico Zio. Genetic algorithms and monte carlo simulation for optimal plant design. *Reliability Engineering & System Safety*, 68(1):29–38, 2000.

[5] Celso MF Lapa, Cláudio MNA Pereira, and Antônio Carlos de A Mol. Maximization of a nuclear system availability through maintenance scheduling optimization using a genetic algorithm. *Nuclear Engineering and Design*, 196(2):219–231, 2000.

[6] Jorge Luiz C Chapot, Fernando Carvalho Da Silva, and Roberto Schirru. A new approach to the use of genetic algorithms to solve the pressurized water reactor's fuel management optimization problem. *Annals of Nuclear Energy*, 26(7):641–655, 1999.

[7] Ryota Omori, Yasushi Sakakibara, and Atsuyuki Suzuki. Applications of genetic algorithms to optimization problems in the solvent extraction process for

spent nuclear fuel. *Nuclear technology*, 118(1):26–31, 1997.

[8] Derek Bingham, Pritam Ranjan, and William J Welch. Design of computer experiments for optimization, estimation of function contours, and related objectives. *Statistics in Action: A Canadian Outlook*, page 109-124, 2014.

[9] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[10] Sam R Thangiah, Ibrahim H Osman, and Tong Sun. Hybrid genetic algorithm, simulated annealing and tabu search methods for vehicle routing problems with time windows. *Computer Science Department, Slippery Rock University, Technical Report SRU CpSc-TR-94-27*, 69, 1994.

[11] WD Li, SK Ong, and AYC Nee. Hybrid genetic algorithm and simulated annealing approach for the optimization of process plans for prismatic parts. *International Journal of Production Research*, 40(8):1899–1922, 2002.

[12] Po-Han Chen and Seyed Mohsen Shahandashti. Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints. *Automation in Construction*, 18(4):434–443, 2009.

[13] Younis Elhaddad and Omar Sallabi. A new hybrid genetic and simulated annealing algorithm to solve the traveling salesman problem. In *Proceedings of the World Congress on Engineering*, volume 1, pages 11–14, 2010.

[14] James Paul Holloway, Derek Bingham, Chuan-Chih Chou, Forrest Doss, R Paul Drake, Bruce Fryxell, Michael Grosskopf, Bart Van der Holst, Bani K Mallick, Ryan McClarren, et al. Predictive modeling of a radiative shock system. *Reliability Engineering & System Safety*, 96(9):1184–1193, 2011.

[15] Carla Currin, Toby Mitchell, Max Morris, and Don Ylvisaker. Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association*, 86(416):953–963, 1991.

[16] Dave Higdon, Marc Kennedy, James C Cavendish, John A Cafeo, and Robert D Ryne. Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing*, 26(2):448–466, 2004.

[17] Marc C Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

[18] Bin Liu and Chunlin Ji. Automated metamaterial design with computer model emulation and bayesian optimization. In *Applied Mechanics and Materials*, volume 575, pages 201–205, 2014.

[19] Fred Glover. Tabu search-part i. *ORSA Journal on computing*, 1(3):190–206, 1989.

[20] Sergiy Butenko. Course: ISEN-621 class lectures, Spring:2015, Texas A&M University, College Station, TX, 77843.

[21] Fred Glover. Tabu search: A tutorial. *Interfaces*, 20(4):74–94, 1990.

[22] Charles Darwin. On the origins of species by means of natural selection. *London: Murray*, page 247, 1859.

[23] James Kennedy. Particle swarm optimization. In *Encyclopedia of Machine Learning*, pages 760–766. Springer, New York, NY, US, 2010.

[24] Riccardo Poli, James Kennedy, and Tim Blackwell. Particle swarm optimization. *Swarm intelligence*, 1(1):33–57, 2007.

[25] Russ C Eberhart and James Kennedy. A new optimizer using particle swarm theory. In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, volume 1, pages 39–43. New York, NY, 1995.

[26] Marco Dorigo, Mauro Birattari, and Thomas Stützle. Ant colony optimization. *Computational Intelligence Magazine, IEEE*, 1(4):28–39, 2006.

[27] David E Golberg. Genetic algorithms in search, optimization, and machine learning. *Addion wesley*, 1989, 1989.

[28] Lawrence Davis. *Handbook of genetic algorithms.* Van No Strand Reinhold, New York, 1991.

[29] Scott Kirkpatrick, C Daniel Gelatt, Mario P Vecchi, et al. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[30] Carl Edward Rasmussen. *Gaussian processes for machine learning.* Springer, Heidelberg, 2006.

[31] Jaakko Leppnen. Serpenta continuous-energy monte carlo reactor physics burnup calculation code. *VTT Technical Research Centre of Finland*, 4, 2013.

[32] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.

[33] Yacov Y Haimes, LS Ladson, and David A Wismer. Bicriterion formulation of problems of integrated system identification and system optimization, *IEEE Transactions on Systems Man and Cybernetics* 3(1971): 296.

[34] Jon C Helton and Freddie Joe Davis. Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliability Engineering & System Safety*, 81(1):23–69, 2003.

[35] RL Iman, JM Davenport, and DK Zeigler. Latin hypercube sampling technical report sand79-1473. *Sandia Laboratories. Albuquerque*, 1980.

[36] S. Surjanovic and D. Bingham. Virtual library of simulation experiments: Test functions and datasets. Retrieved July 25, 2016, from http://www.sfu.ca/ ssurjano.

[37] Ernesto P Adorio and U Diliman. Mvf-multivariate test functions library in c for unconstrained global optimization. *Quezon City, Metro Manila, Philippines*, 2005.

[38] Marcin Molga and Czesław Smutnicki. Test functions for optimization needs. *Test functions for optimization needs*, 2005.

[39] Akansha Kumar, Sunil S Chirayath, and Pavel V Tsvetkov. Analysis of a sustainable gas cooled fast breeder reactor concept. *Annals of Nuclear Energy*, 69:252–259, 2014.

[40] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2951–2959, 2012.

[41] Alexander J Smola, Peter Bartlett, et al. Sparse greedy gaussian process regression. *Advances in Neural Information Processing Systems*, 13:619–625, 2001.

[42] Carl Edward Rasmussen, JM Bernardo, MJ Bayarri, JO Berger, AP Dawid, D Heckerman, AFM Smith, and M West. Gaussian processes to speed up hybrid monte carlo for expensive bayesian integrals. In *Bayesian Statistics 7*, pages 651–659, 2003.

[43] Akio Yamamoto. A quantitative comparison of loading pattern optimization methods for in-core fuel management of pwr. *Journal of Nuclear Science and*

Technology, 34(4):339–347, 1997.

[44] JG Stevens, KS Smith, KR Rempe, and TJ Downar. Optimization of pressurized water reactor shuffling by simulated annealing with heuristics. *Nuclear Science and Engineering*, 121(1):67–88, 1995.

[45] N Zavaljevski. A model for fuel shuffling and burnable absorbers optimization in low leakage pwrs. *Annals of Nuclear Energy*, 17(4):217–220, 1990.

[46] Zhiwen Xu. *Design strategies for optimizing high burnup fuel in pressurized water reactors.* PhD dissertation, Massachusetts Institute of Technology, Cambridge, MA , USA 2003.

[47] Robert J Fetterman. Advanced first core design for the westinghouse ap1000. In *17th International Conference on Nuclear Engineering*, pages 167–174. American Society of Mechanical Engineers, 2009.

[48] Clifford J Stanley and Frances M Marshall. Advanced test reactor: A national scientific user facility. In *16th International Conference on Nuclear Engineering*, pages 367–372. American Society of Mechanical Engineers, 2008.

[49] AM Savchenko, AV Vatulin, VI Sorokin, GV Kulakov, SV Maranchak, and KV Lipkina. New concept of designing composite fuel for fast reactors with closing fuel cycle. In *Fast Reactors and Related Fuel Cycles: Safe Technologies and Sustainable Scenarios (FR13). companion cd-rom. Proceedings of an International Conference*, 2015.

[50] AM Savchenko, II Konovalov, AV Vatulin, and EM Glagovsky. New concept of designing pu and ma containing fuel for fast reactors. *Journal of Nuclear Materials*, 385(1):148–152, 2009.

[51] Edward Lloyd Snelson. *Flexible and efficient Gaussian process models for machine learning*. PhD dissertation, University of London, London, UK, 2007.

[52] Ercan Solak, Roderick Murray-Smith, William E Leithead, Douglas J Leith, and Carl Edward Rasmussen. Derivative observations in Gaussian process models of dynamic systems. *Conference on Neural Information Processing Systems*, Vancouver, Canada, 9-14 December 2002.

[53] Roderick Murray-Smith, Tor A Johansen, and Robert Shorten. On transient dynamics, off-equilibrium behaviour and identification in blended multiple model structures. Control Conference (ECC), 1999 European. IEEE, 1999.

[54] Tadahiko Murata, Hisao Ishibuchi, and Hideo Tanaka. Genetic algorithms for flowshop scheduling problems. *Computers & Industrial Engineering*, 30(4):1061–1071, 1996.

[55] Il-Kwon Jeong and Ju-Jang Lee. Adaptive simulated annealing genetic algorithm for system identification. *Engineering Applications of Artificial Intelligence*, 9(5):523–532, 1996.

[56] Jarno Vanhatalo, Jaakko Riihimäki, Jouni Hartikainen, Pasi Jylänki, Ville Tolvanen, and Aki Vehtari. Gpstuff: Bayesian modeling with gaussian processes. *The Journal of Machine Learning Research*, 14(1):1175–1179, 2013.

[57] R Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, 2012.

[58] Laurie S Waters et al. Mcnpx users manual, Los Alamos, NM. Accessed in Apr 15, 2012 at http://mcnpx.lanl.gov/opendocs/versions/v230/MCNPX_2.3.0_Manual.pdf, 2002.

[59] Forrest B Brown et al. Mcnp–a general monte carlo n-particle transport code, version 5. Los Alamos National Laboratory, Oak Ridge, TN, 2003.

[60] JP Segala. Final safety evaluation report related to certification of the AP1000 standard design (NUREG-1793). *America, Nuclear Regulatory Commission*, 2004.