

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The basic objective of economic load dispatch (ELD) of electric power generation is to schedule the committed generating unit outputs so as to meet the load demand at minimum operating cost while satisfying all unit and system equality and inequality constraints. In addition, the increasing public awareness of the environmental protection and the passage of the Clean Air Act Amendments of 1990 have forced the utilities to modify their design or operational strategies to reduce pollution and atmospheric emissions of the power plants. This makes the ELD problem a large scale highly nonlinear constrained optimization problem. Also, these objectives are conflicting in nature and cannot be handled by conventional single objective optimization techniques. Single objective optimization techniques give optimal solution in respect of a single aspect, i.e. they give the best value of the objective function under consideration. The values of other objectives at such a solution may be optimum or not. The way out, therefore, lies in the multiobjective approach to problem solving.

Multiobjective optimization (or programming), also known as multi-criteria or multi-attribute optimization, is the process of simultaneously optimizing two or more conflicting objectives subjected to certain constraints. The solution of multiobjective optimization gives us a number of solutions called **noninferior solutions**. The objectives considered for optimal power dispatch are—cost of generation (F_C), system transmission losses (F_L) and environmental pollution (F_P). In economic load dispatch, cost of generation is considered as the objective function to be minimized while satisfying the load demand and losses.

A feasible solution to a Multiobjective programming problem is noninferior if there exists no other feasible solution that will yield an improvement in one objective without causing degradation in at least one of the other objectives. A given noninferior solution may or may not be acceptable to the decision maker. However, it is important to note that, it is one of the noninferior solutions for which decision maker looks for.

The ideal situation where one would like to operate the power systems is one where all the objectives are minimum. But this is not feasible due to conflicting nature of objectives. Therefore, one can achieve a point which is noninferior and at the minimum distance from the Ideal Point. Such a point is known as the **Target Point (TP)** or the best compromise solution.

There are various techniques for generating noninferior solutions - weighting method[1], constraint method and NISE method[2,3] etc. In this thesis; the Multiobjective Economic Load Dispatch (MOELD)[4] problem has been formulated using weighting method and has been solved by GA tool of MATLAB. This gives us noninferior solutions in 3D space for IEEE 5 Bus system. The distance of all the feasible operating points (noninferior solutions) from the Ideal power system operation point is calculated by Minimum Distance Method [5] and the optimal power system operation is one for which this distance is minimum. This method directly gives the best compromise solution.

1.2 OBJECTIVES AND METHODOLOGY

Our objective is to solve Economic Load Dispatch in 3D space i.e considering three objectives of power system- cost of generation, system transmission losses and environmental pollution. The Multiobjective Economic Load Dispatch (MOELD) problem or Economic Load Dispatch in 3D space has been formulated using weighting method for IEEE 5 bus system. GA toolbox of MATLAB has been used to generate the noninferior set. The Target Point (TP) has been achieved by minimum distance method technique [5].

1.3 LITERATURE SURVEY

1.3.1 Genetic Algorithm

GA is a global search algorithm based on biological concepts which mimic the mechanics of nature and natural genetics. Along with Evolutionary Programming, Evolutionary Strategy, and Genetic Programming, GA is a part of a wider concept called Evolutionary Computation (EC). Meanwhile, EC, along with Adaptive-Neural Networks (ANN), Fuzzy Systems, amongst others,

are classified as Artificial Intelligence (AI) techniques [6]. GA does not require derivative information or other auxiliary knowledge, except objective or fitness functions. GA is capable of finding the global optimum and of coping with various difficulties, such as non-linearity, nonsmoothness, discontinuity, and non-convex characteristics [7, 8].

1.3.1.1 Various GA Techniques:- Although GA methods have the same basic principles there are a wide-range of techniques that can be used to look for the most effective and efficient solutions. Many authors have used different techniques in the application of the Economic Load Dispatch (ELD) problem to seek the most effective technique for solving various problems.

Encoding/decoding techniques: - Among the early work is a paper by Walters and Sheble [9]. In this paper, they encoded generator output values into binary strings and investigated two types of binary encoding. Another unique encoding method is offered by Kumaran and Mouly [10]. Besides binary coded GA, some work has been done based on real coded GA (RCGA) for different Economic Load Dispatch (ELD) problems with satisfactory results. Chiang [11], Zhang et al [12], Wong and Wong [13], Abido [4], and Das and Patvardhan [14] use RCGA to solve valve-point loading problem. Abido [4] and Das and Patvardhan [14] use GA in a multiobjective optimization problem. From their work, it is shown that RCGA is an effective technique for various scenarios and has the capability of being combined with other methods.

Objective and Constraint Function Handling:- An objective function in GA is transformed into a fitness function. As for constraint functions, if possible they are satisfied in the population construction, such as the minimum and maximum operating limits. The highest encoding value represents the maximum operation limit, and the lowest value for minimum limit. If this technique is impossible or ineffective, the other common technique is to handle the constraint function by including it in the fitness function, along with the objective function. Hence, the fitness function will represent two purposes at the same time, i.e. optimizing the objective value and satisfying the constraints. A simple example is a fitness function formulated by Kumaran and Mouly [10], which sums all of the function representing cost, load balance and loss objectives. In this technique each part of the fitness function is equally weighted.

1.3.1.2 GA Operators:- Typically, GA uses crossover and mutation as operators for producing individuals for the subsequent generations; therefore all authors use these operators in their

papers. The probability of crossover (P_c) is usually high, whereas the probability of mutation (P_m) is always very low. These probabilities reflect what happens in nature, where probability of crossover is high and probability of mutation is low. The values for P_c and P_m are chosen so as to find a suitable balance between fast convergence and increasing the diversity of the population. Another GA operator used in some papers (Chiang [11], and Chiang et al [15]) is migration. This operator is applied to increase the diversity of the population after a pre-specified generation by generating newly diverse individuals of a small part of the population in the space search.

1.3.1.3 The Size of Population and the Number of Generations:- The size of population and the number of generations used in the papers vary widely depending on techniques used, as well as the size and complexity of system modeled. Walters and Sheble [9] as well as Sheble and Brittig [8] utilise 100 chromosomes and 100 generations for a small system with 3 generator units, on the other hand Chen and Chang [16] only use 16 chromosomes and around 20 generations for a large system with 40 generators. For Environmental Economic Power Dispatch problem, Abido [4] selected the size of population and the number of generations as 200 and 500 respectively. Nanda and Narayanan [17] investigate three different population sizes (10, 15, and 20) and three different numbers of generations (5, 10, and 15) for the same systems and assert that in this case the population size of 10 with 15 generations provides an optimum solution. In a unique piece of work by Wong and Wong [13] only two chromosomes are used in each generation, but they produce 40 chromosomes from crossover.

1.3.1.4 Hybrids of GA with Other methods:- Besides the simplicity of the procedure, GA methods can be improved and easily combined with other methods creating a hybrid GA. In an early work a hybrid GA was developed by Wong and Wong [13] who investigated a hybrid of GA and Simulated Annealing, called Genetic Annealing Algorithm (GAA). A different combination technique proposed by Ongsakul and Ruangpayoongsak [18] is a Genetic Algorithm based on a Simulated Annealing solution (GA-SA). Their algorithm is relatively simple, where both SA and GA are used in sequence. The results are compared with some other methods, including dynamic programming (DP), Simulated Annealing (SA), merit order loading, and local search. Integrating GA and a Tabu Search (TS) technique is done by Sudhakaran and Slochanal [19] for the system with combined heat and power economic dispatch. TS are characterized by the capability to avoid local optima traps by memorizing a short set of recent

solutions. Kumarappan and Mohan [20] proposed a neuro-hybrid GA method for solving ELD, which consists of three methods, i.e. Artificial Neural Network (ANN), TS and GA. The hybrid of GA with fuzzy logic controller (FCGA) is studied by Wang et al [21]. They use a fuzzy logic controller in the crossover and mutation processes to improve their results by dynamically modifying the crossover and mutation rate during the process.

1.3.2 Multiobjective Optimization

Optimization refers to finding the best possible solution to a problem given a set of limitations (or constraints). When dealing with a single objective to be optimized (e.g. the cost of a design), we aim to find the best possible solution available (called “global optimum”), or at least a good approximation of it. However, when devising optimization models for a problem, it is frequently the case that there is not one but several objectives that we would like to optimize. In fact, it is normally the case that these objectives are in conflict with each other. These problems with two or more objective functions are called “multi-objective” and require different mathematical and algorithmic tools than those adopted to solve single objective optimization problems.

A Multiobjective Problem (MOP) is a problem which has two or more objectives that are to be optimized simultaneously along with constraints imposed on the objectives.

Most MOPs, do not lend themselves to a single solution and have, instead, a set of solutions. Such solutions are really “trade-offs” or good compromises among the objectives. In order to generate these trade-off solutions, an old notion of optimality is normally adopted. This notion of optimality was originally introduced by Francis Ysidro Edgeworth in 1881 [22] and later generalized by Vilfredo Pareto in 1896 [23].

The Operations Research community has developed approaches to solve MOPs since the 1950s. A number of mathematical programming techniques have been developed to solve MOPs [24, 25]. Coello and Aguirre [26] proposed that the constraints of a single objective problem be handled as objectives. Whereas Jensen [27] and Knowles et al [28] proposed conversion of single objective optimization into ‘multiobjectivization’ and Reduction of Local Optima in single objective problems by multiobjectivization respectively. Kennedy and Eberhart [29] proposed alternative bio-inspired heuristic called Particle Swarm Optimization. Price [30] introduced Differential Evolution for solving multiobjective problems. These techniques were used by Abbas

and Sarker [31] and Coello et.al [32] to solve multiobjective problems. Coello and Cruz [33] used artificial immune system to solve multiobjective optimization problem. Guntsch [34] used Ant Colony Optimization to solve stochastic problems.

To solve MOPs, initially parameters are set which are further fine tuned by hand. Despite of this tuning by hand, design of self adaptation techniques [35, 36] are new area of for research.

1.3.3 Economic Load Dispatch

Economic load dispatch (ELD) is one of the major issues in power system operation [37]. It is defined as a process of allocating the output of generators to satisfy electrical demand in a power system in the most economic way considering all constraints [38]. The complexity of the ELD problem depends upon many factors, such as the size of the system, system constraints, and generator characteristics.

Several techniques have been introduced to solve the optimization of ELD, which can be divided into conventional and stochastic methods. Conventional methods use a deterministic approach, such as the LaGrange multiplier [39], Linear Programming (LP) [40] and Dynamic Programming (DP) [41]. These methods have limitations or drawbacks when coping with more complex problems.

Recent techniques have been developed using stochastic approaches for solving optimization problems. Examples are an Adaptive Hopfield Neural Network [42], the Simulated Annealing method [18] and Genetic Algorithms (GA)[43], amongst others. These new methods offer alternative techniques which attempt to overcome the drawbacks of conventional methods.

The GA method has been used for solving various power system architectures in terms of size, generation characteristics, system constraints, or objective functions by many authors. This shows the flexibility and capability of the GA method to solve ELD.

Amongst the first work, Sheble and Brittig [8] examined GA to satisfy typical smooth quadratic functions for three thermal generators that can also be solved using the classical LaGrange technique. They used the fact that GA can provide similar results with the classical solution to validate the effectiveness of GA.

In a recent study, Chiang [15] reports the use of GA for another complex ELD problem that deals with valve point loading and prohibited operating zones (POZ). Using simulation examples, he asserts that his proposed GA method has many merits, such as being straightforward, easy to implement, and more effective.

Hong and Li [43] study the effectiveness of using GA for a system consisting of multiple cogenerators and multiple buyers in a deregulated market. They successfully use GA in both an IEEE 30-bus system and an IEEE 118-bus system.

Hosseini and Kheradmandi [44] use a GA method in a deregulated power system which considers transmission costs and ramping rate constraints. They successfully test their GA method both on a 10-unit system and an IEEE 30-bus system. Abido [4] proposed a novel approach based on GA for solving ELD which considers environmental objectives. The problem is formulated into a multiobjective optimization problem with competing fuel cost objective and emission cost minimization. His proposed GA method provides a representative and manageable noninferior set.

Hong and Li [45] report on using GA for short-term scheduling of an autonomous system containing diesel generators, wind power, solar photovoltaics and batteries. The result is compared to Simulated Annealing (SA) for the same problem and provides a solution that requires fewer iterations and takes less time.

Chen and Chang [16] used GA for a large-scale system in Taiwan Power System which contains 40 units, taking into account transmission losses, ramp rate limits and prohibited zones as well. They report the robustness and powerfulness of GA compared to Lambda Iteration Methods for solving this problem.

In order to provide a better optimal set in a multiobjective ELD problem, Abido [4] employs hierarchical clustering and a fuzzy base mechanism into the GA procedure. The hierarchical clustering is used to reduce the number of optimal sets, without destroying the tradeoff characteristics between objectives. A fuzzy-based mechanism is applied at the end to find out the best compromise solution.

Many papers use penalty factors for solving constraint problems, such as the papers by Hong and Li [45], Chiang [11], Chiang et al [15], and Nanda and Narayan [17]. Hosseini and Kheradmandi [44] do not use penalty factors, but they set the objective function to a specific large value if the solutions do not satisfy the constraints. Otherwise, they do not change the objective function value. There are two basic techniques for solving multiobjective ELD problems. The first is to convert it into one objective function, which usually gives the best solution. Using this technique, Ma et al [46] convert the emission objective into a cost function and Kumaran and Mouly [10] convert the minimization of losses and the cost objective, along with the load balance constraint, into an index value and then all index values are summed into a single fitness function. The second technique is to use a specific multiobjective method. In this technique, all objective functions are in competition and a search algorithm is used to find an optimal solution. In noninferior set, a solution cannot be improved upon without adversely affecting the other objectives. Therefore, the result will be a set of optimal solutions that can be presented in a trade-off curve among all objectives. The second approach is used by Abido [4] and Yalcinoz [47].

1.4 PLAN OF THESIS

This dissertation has been arranged in seven chapters. The contents of the chapters are briefly outlined as indicated below:

Chapter 1: Discusses the introduction to research objectives of the thesis. Literature survey of the covered topics has also been presented.

Chapter 2: Presents Genetic Algorithms and its applications.

Chapter 3: Minimization of Rosenbrock function is carried out using genetic algorithm.

Chapter 4: Discusses the Multiobjective Optimization. This presents formulation of general multiobjective optimization problem and the concept of Noninferiority, Weighing method and Ideal distance minimization method.

Chapter 5: Discusses Multiobjective Approach to Economic Load Dispatch and deals with problem formulation in 3D space for IEEE 5 bus system.

Chapter 6: Results have been presented and Ideal distance minimization method has been applied for finding the Target Point.

Chapter 7: Conclusion and the prospects for Future Directions have been discussed.

Appendix and references are at the end of the thesis.

CHAPTER 2

GENETIC ALGORITHM

2.1 INTRODUCTION

In engineering disciplines a large spectrum of optimization problems have grown in size and complexity. In some instances, the solution for complex multidimensional problems by using classical optimization technique is difficult or expensive. This realization has led to an increased interest in a special class of searching algorithm, namely, evolutionary algorithm (EA) [46, 47] and their foundations lie in the evolutionary patterns observed in living beings.

In this area of operational research, there exist several primary branches

1. Genetic algorithm(GA)
2. Evolutionary programming (EP)
3. Evolutionary strategies(ES)

To date GA is the most widely known technology. The optimization technique has been applied to many complex problems in the fields of industrial and operational engineering. In power system, well known applications include unit commitment, economic dispatch, load forecasting, reliability studies and various resource allocation problems.

2.1.1 General Structure of GA

As stochastic search typical structure of GAs was described by Goldberg [48]. Essentially, GAs are referred to as stochastic search techniques that are based on the Darwinian thinking of natural selection and natural genetics. In general GAs start with an initial set of random solutions that lie in the feasible solution space. This random cluster of solution point is called the population. Each solution in the population represents a possible solution to the optimization problem therefore called the chromosome. The chromosome is a string of symbols based on the uniqueness of two state machines; they are commonly binary bit string.

2.2 DEFINITIONS AND CONCEPTS USED IN GENETIC COMPUTATION

GAs have their foundations both in natural biological genetics and in modern computer science (Table 2.1). As such, nomenclature used in this, inherently a mix of both natural and artificial intelligence.

To understand the roots of GAs, we look at biological analogy. In biological organisms, a **chromosome** carries a unique set of information that encodes the data on how the organism is constructed. A collection or complete set of chromosome is called **phenotype**. Also, within each chromosome are various individual structures called **genes**, which are specific coded features of organisms. The possibility of the genes for one trait is called **allele** and unique position of every gene on the chromosome is called **locus**. **Genotype** is a group of organisms with the same genetic constitution.

With the basic understanding, the following terminologies and concepts are summarized.

Table 2.1 Terminology in Genetic Algorithms

GA Terms	Corresponding Optimization Description
Chromosomes	Solution set
Gene	Part of solution
Alleles	Value of gene
Phenotype	Decoded solution
Genotype	Encoded solution
Locus	Position of gene

2.2.1 Evolutionary Algorithms:

EAs represent a broad class of computer based problem solving systems. Their key feature is the evolutionary mechanisms that are at the root of formulation and implementation. Of course, EAs by themselves represent a special class of new intelligent system (IS) used in many global optimization algorithms. Fig. 2.1 shows the various categories of IS and the position of the GA as one of the more commonly known EP techniques [49, 50].

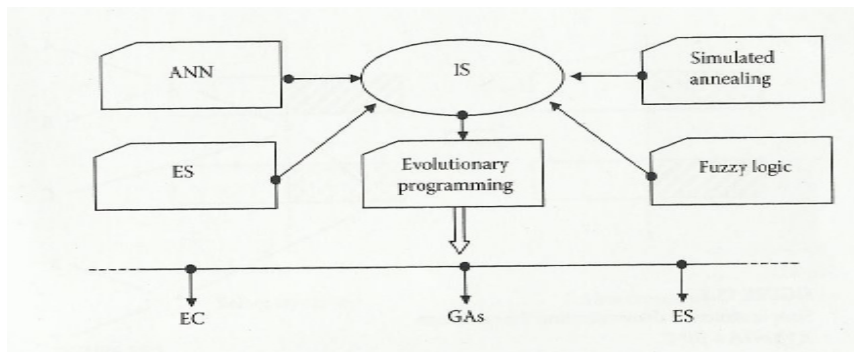


Fig 2.1 Common Classifications of IS

2.3 GA APPROACH

GAs are general purpose search techniques based on principles inspired by the genetic evolutionary mechanisms observed in the populations of natural systems and living organisms. Typically there are several stages in the optimization process:

Stage 1: Creating an initial population.

Stage 2: Evaluating the fitness function.

Stage 3: Creating new populations.

2.3.1 GA Operators:

Various operators are used to perform the tasks of the stages in a GA: The production or elitism operator, crossover operator, and the mutation operator. The production operator is responsible for generating any copies of individuals that satisfy the goal function. That is, they either pass the fitness test of goal function or otherwise are eliminated from the solution space. The crossover operator is used for recombination of individuals within the generation. The operator selects two individuals within the current generation and performs swapping at a random or fixed site in the individual string (Fig 2.2). The objective of the crossover process is to synthesize bits of knowledge from the parent chromosomes that will exhibit improved performance in the offspring.

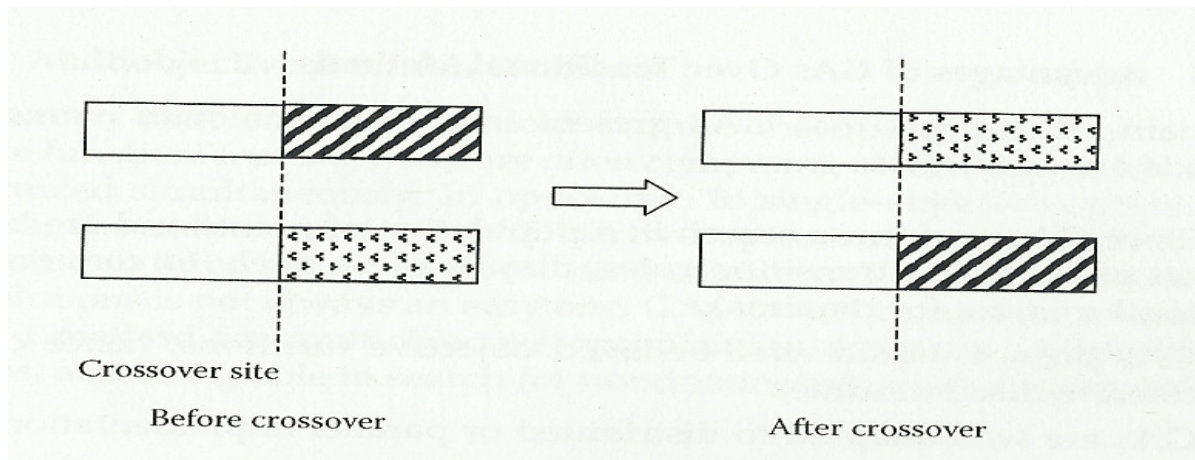


Fig 2.2 Crossover operation on a pair of strings

The certainty of producing better performing offspring via the crossover process is one important advantage of GAs.

Finally, the mutation operator is used as an exploratory mechanism that aids the requirements of finding a global extrema to the optimization problem. Basically, it is used to randomly explore the solution space by flipping bits of selected chromosomes or candidates from the population. There is an obvious trade-off in the probability assigned to the mutation operator. If the frequency were high, the GA would result in completely random search with a large loss of data integrity. On the other hand, too low frequency assigned to this operator may result in an incomplete scan of the solution space.

2.3.2 Major Advantages:

GAs have received considerable attention regarding their potential as a novel optimization technique. There are several major advantages when applying GAs to optimization problems.

1. GAs do not have many mathematical requirements for optimization problems. Due to their evolutionary nature, GAs will search for solutions without regard to specific inner workings of the problem. They can handle any kind of objective functions, and any kind of constraints (i.e. linear or non-linear) defined on discrete, continuous, or mixed search spaces.
2. Ergodicity of evolution operators makes GAs very effective at performing global searches (in probability). The traditional approaches perform local search by a convergent stepwise procedure, which compares the values of nearby points and moves to the relative optimal points. Global optima can be found only if the problem possesses certain convexity properties that essentially guarantee that any local optima is a global optima.
3. GAs provide us with a great flexibility to hybridize with domain dependent heuristics to make an efficient implementation for a specific problem.

2.3.3 Advantages of GAs over Traditional Methods:

The main advantages that present in comparison with conventional methods are as follows:

1. Since GAs perform a search in a population of points and are based on probabilistic transition rules, they are less likely to converge to local minima (or maxima).
2. GAs do not require well-behaved objective functions, hence easily tolerate discontinuities.
3. GAs are well adapted to distributed or parallel implementations.
4. GAs code parameters in a bit string and not in the values of parameters. The meaning of the bits is completely transparent for the GA.
5. GAs search from a population of points and not from a single point.

6. GAs use transition probabilistic rules (represented by the selection, crossover, and mutation operators) instead of deterministic rules.

Nevertheless, the powers of conventional methods are recognized. The GA should only be used when it is impossible (or very difficult) to obtain efficient solutions by these traditional approaches.

2.4 THEORY OF GAs

2.4.1 Constraints

Most optimization problems are constrained in some way. GAs can handle constraints in two ways, the most efficient of which is by embedding these in the coding of chromosomes. When this is not possible, the performance of invalid individuals should be calculated according to a penalty function, which ensures that these individuals are, indeed, poor performers. Appropriate penalty functions for a particular problem are not necessarily easy to design, since they may considerably affect the efficiency of the genetic search.

2.4.2 Other GA Variants

The simple GA has been improved in several ways. Different selection methods have been proposed [48] that reduce the stochastic errors associated with roulette wheel selection. Ranking has been introduced as an alternative to proportional fitness assignment, and has been shown to help avoidance of premature convergence and to speed up the search when the population approaches convergence. Other recombination operators have been proposed, such as the multiple point and reduce-surrogate crossover. The mutation operator has remained more or less unaltered, but the use of real-coded chromosomes require alternative mutation operator, such as intermediate crossover. Also, several models of parallel GAs have been proposed, improving the performance and allowing the implementation of concepts such as that of genetic isolation. This method works well with bit string representation. The performance of GAs depends on the performance of the crossover operator used.

The crossover rate P_c is defined as the ratio of the number of offsprings produced in each generation to the population size (denoted pop size). A higher crossover rate allows exploration of more of the solution space and reduces the chances of settling for false optimum; but if this rate is too high, a lot of computational time will be wasted.

Mutation is a background operator that produces spontaneous random changes in various chromosomes. A simple way to achieve mutation would be to change one or more genes.

The mutation rate P_m is defined as the percentage of the total number of genes replaced in the population and it controls the rate at which new genes are introduced into the population for trial. If it is too low, many genes that would have been useful are never tried. But if it is too high, there will be many random populations, the offspring will start losing their resemblance to the parents, and the algorithm will lose the ability to learn from the history of the search.

2.4.3 Coding

Each chromosome represents a potential solution for the problem and must be expressed in binary form in the integer interval 0-21. We could simply code any integer in binary base, using four bits (such as 1001 or 0101). If we have a set of binary variables, a bit will represent each variable. For a multivariable problem, each variable has to be coded in the chromosome.

2.4.4 Fitness

Each solution must be evaluated by a function to produce a specific value. This objective function is used to model and characterize the problem to be solved. In many instances, the fitness function can be simulated as the objective function used in classical optimization problems. In such cases, these optimization problems may be unconstrained or constrained. For the latter case, a Lagrangian or penalty approach can be used in formulating a suitable fitness function.

Notably, the fitness function does not necessarily have to be in closed mathematical form. It can also be expressed in quantitative form and, in power system applications, with fuzzy models.

2.4.5 Selection

The selection operator creates new populations or generations by selecting individuals from the old population. The selection is probabilistic but biased towards the best as special deterministic rules are used. In the new generations created by the selection operator, there will be more copies of the best individual and fewer copies of the worst. Two common techniques for implementing the selection operator are the stochastic tournament and roulette wheel approaches [48]. We have used the Stochastic tournament approach.

1. Stochastic tournament: This implementation is suited to distributed implementations and is very simple: every time we want to select an individual for reproduction, we choose two, at random, and the best wins with some fixed probability, typically 0.8. This scheme can be enhanced by using more individuals in the competition or even considering evolving winning probability.
2. Roulette wheel: In this process, the individuals of each generation are selected for survival into the next generation according to a probability value, proportional to the ratio of individual fitness over total population fitness; this means that on an average the next generation will receive copies of an individual in proportion to the importance of its fitness value.

We have used the Roulette wheel approach.

2.4.6 Crossover

The basic operator for producing new chromosome is crossover. In this operator, information is exchanged among strings of mating pool to create new strings. The aim of the crossover operator is to search the parameter space. Crossover is a recombination operator, which proceeds in three steps. First, the reproduction operator selects at random a pair of two individual string for mating, then a crossover site is selected at random along the string length and the position values are swapped between two strings following the cross site. There are many types of crossover as Single point crossover, two point crossover, Multipoint crossover, Uniform crossover, Matrix crossover etc. In the single point crossover, two individual strings are selected at random from the mating pool. Next, a crossover site is selected randomly along the string length and binary digits (alleles) are swapped between the two strings at crossover site. Suppose site 3 from left is

selected at random. It means starting from the 4th bit and onwards, bits of strings will be swapped to produce offspring which is given by-

Single point crossover operation

Parent 1: $X_1 = \{ 0\ 1\ 0\ \mathbf{1\ 1\ 0\ 1\ 0\ 1\ 1} \}$

Parent 2: $X_2 = \{ 1\ 0\ 0\ \mathbf{0\ 0\ 1\ 1\ 1\ 0\ 0} \}$

Offspring 1: $X_1 = \{ 0\ 1\ 0\ \mathbf{0\ 0\ 1\ 1\ 1\ 0\ 0} \}$

Offspring 2: $X_2 = \{ 1\ 0\ 0\ \mathbf{1\ 1\ 0\ 1\ 0\ 1\ 1} \}$

In a two point crossover operator, two random sites are chosen and the contents bracketed by these sites are exchanged between two mated parents. If the cross site 1 is three and cross site 2 is six, the strings between three and six are exchanged which is given by-

Two point crossover operation

Parent 1: $X_1 = \{ 0\ 1\ 0\ \mathbf{1\ 1\ 0\ 1\ 0\ 1\ 1} \}$

Parent 2: $X_2 = \{ 1\ 0\ 0\ \mathbf{0\ 0\ 1\ 1\ 1\ 0\ 0} \}$

Offspring 1: $X_1 = \{ 0\ 1\ 0\ \mathbf{0\ 0\ 1\ 1\ 0\ 1\ 1} \}$

Offspring 2: $X_2 = \{ 1\ 0\ 0\ \mathbf{1\ 1\ 0\ 1\ 1\ 0\ 0} \}$

2.4.7 Mutation

The final genetic operator in the algorithm is mutation. In general evolution, mutation is a random process where one allele of a gene is replaced by another to produce a new genetic structure. Mutation is an important operation, because newly created individuals have no new inheritance information and the number of alleles is constantly decreasing. This process results in the contraction of the population to one point, which is wished at the end of convergence process. Diversity is one goal of the learning algorithm to search always in regions not viewed before. Therefore, it is necessary to enlarge the information contained in the population. One way to achieve this goal is **mutation**. The role of mutation is often seen as providing a guarantee that the probability of searching any given string will never be zero and acting as safety net to

recover good genetic material that may be lost through the action of selection and crossover. In GAs mutation is randomly applied with low probability in the range of 0.001 & 0.01 and modifies elements in the chromosome. Here, binary mutation flips the value of the bit at the loci selected to be the mutation point. Given that mutation is applied uniformly to an entire population of strings, it is possible that a given string may be mutated at more than one point.

Mutation operation

Offspring X_1 : 1 1 1 1 0 1 0

New offspring X_2 : 1 1 0 1 0 1 0

2.4.8 Parameters

Like other optimization methods, GAs have certain parameters such as

1. Population size
2. Genetic operations probabilities
3. Number of individuals involved in the selection procedure, and so on

These parameters must be selected with maximum care, for the performance of the GA depends largely on the values used. Normally, the use of a relative low population number, high crossover, and low-mutation probabilities are recommended. Goldberg [48] analyzes the effect of these parameters in the algorithm.

2.5 GENERAL ALGORITHM OF GAs

During successive iterations, called generations, the chromosomes are evaluated, using some measures of fitness. To create the next generations, new chromosomes, called offspring, are formed by either

1. Merging two chromosomes from the current generation using a crossover operator,
or
2. Modifying a chromosomes using a mutation operator

A new generation is formed by

1. Selecting, according to the fitness values, some of the parents and offspring.
2. Rejecting others to keep the population size constant. Chromosomes satisfying various measures of fitness have higher probabilities of being selected.

After several generations, the algorithms converge to the best chromosome, which, it is hoped, represents the optimum or suboptimal solution to the problem.

CHAPTER 3

MINIMIZATION OF ROSENBROCK FUNCTION USING OPTIMTOOL BOX

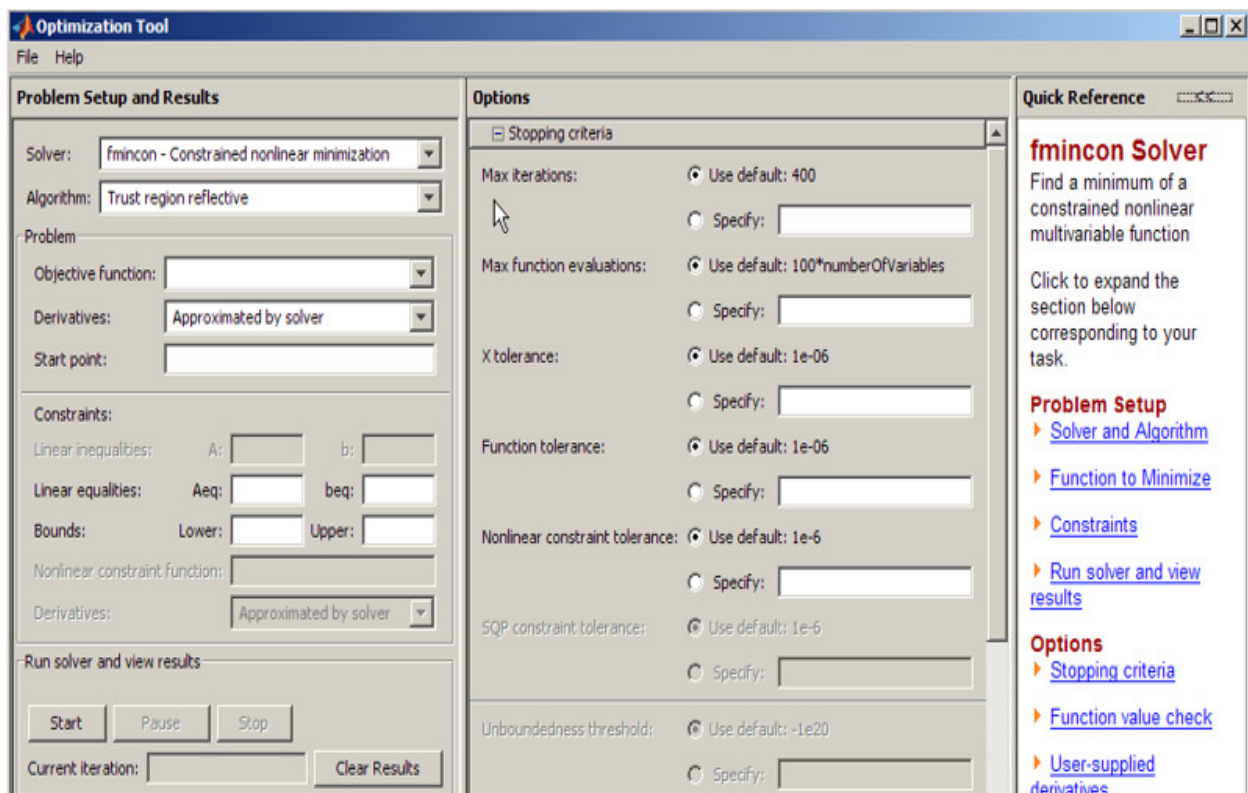
3.1 GETTING STARTED WITH THE OPTIMIZATION TOOL

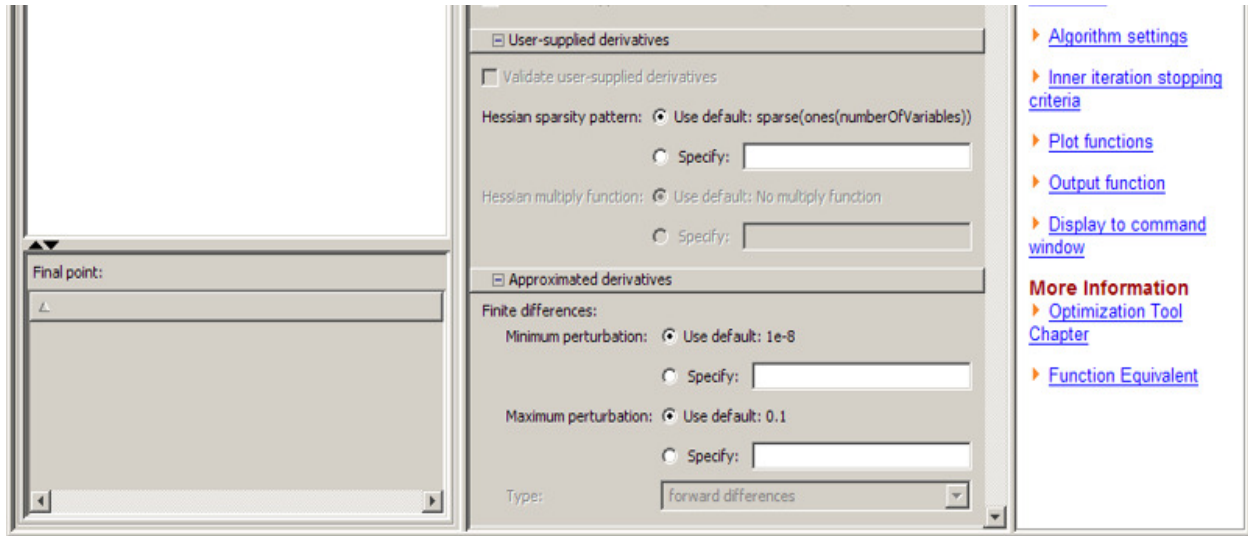
3.1.1 Introduction

The Optimization Tool is a GUI for solving optimization problems. With the Optimization Tool, we select a solver from a list and set up problem visually. We can also import and export data from the MATLAB workspace, and generate M-files containing problem configuration for the solver and options.

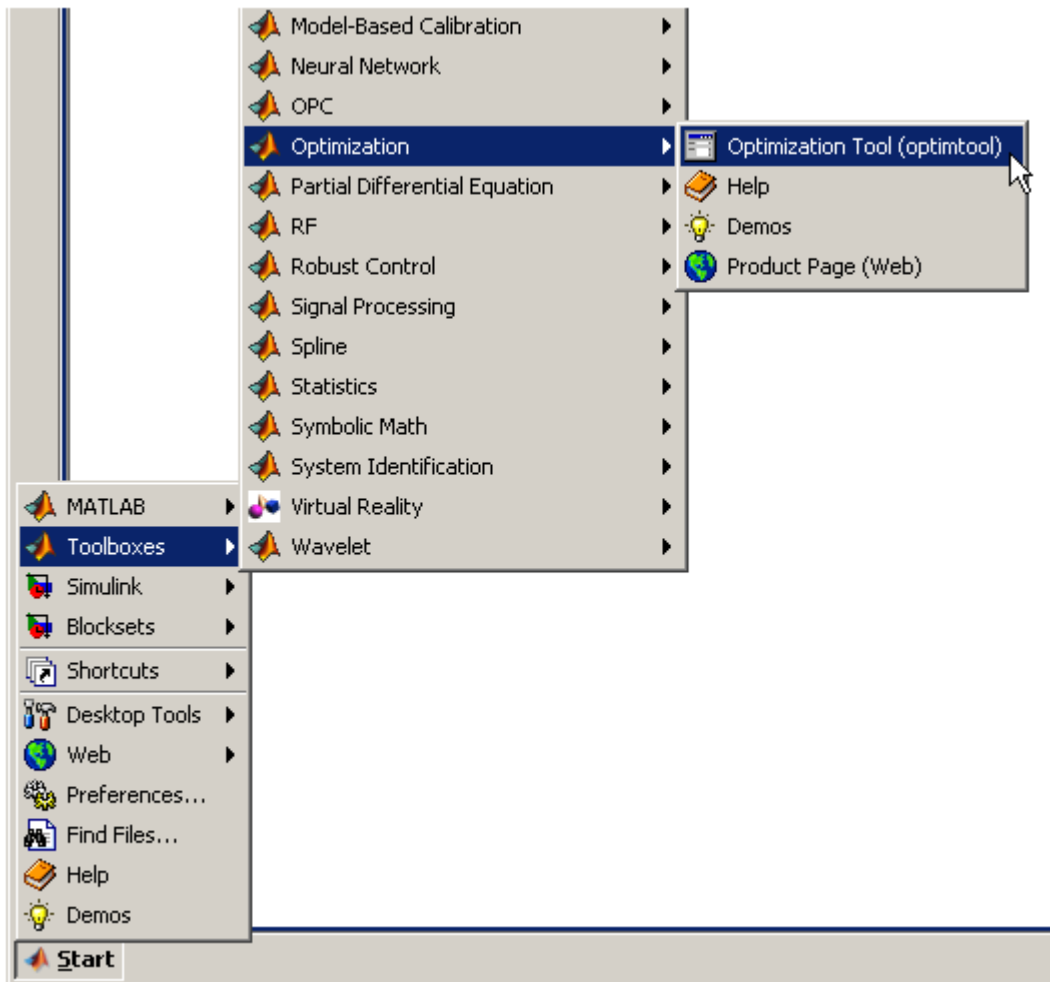
3.1.2 Opening the Optimization Tool

To open the tool, type `optimtool` in the Command Window. This opens the Optimization Tool, as shown in the following figure.



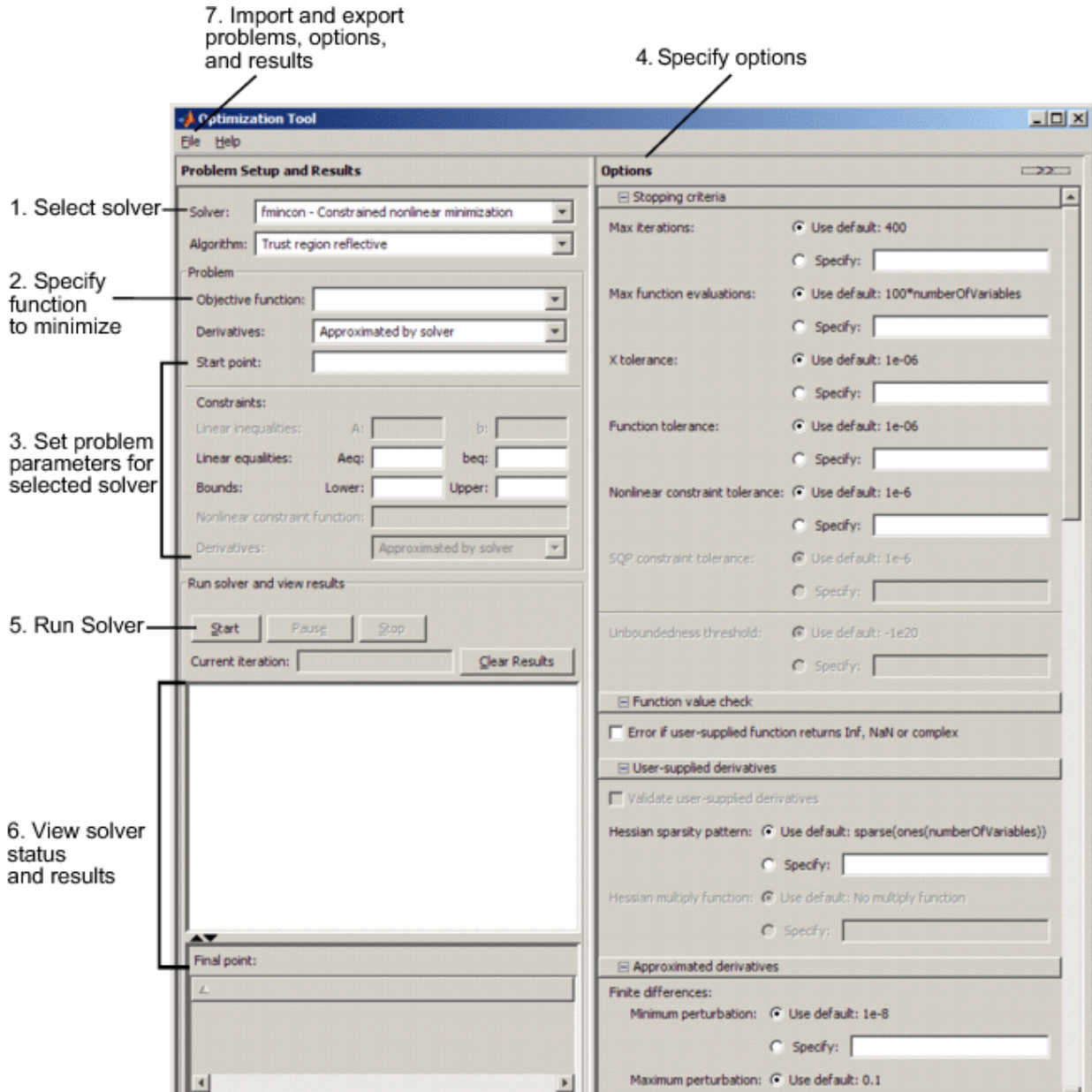


We can also open the Optimization Tool from the main MATLAB window as pictured:



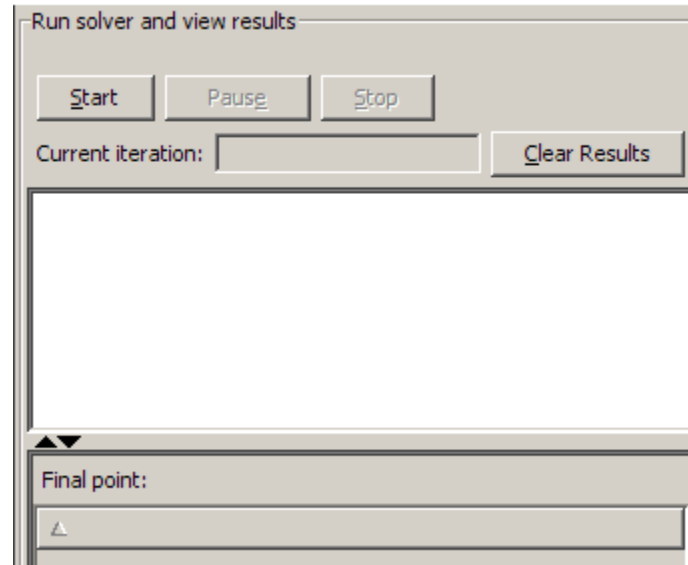
3.1.3 Steps for Using the Optimization Tool

This is a summary of the steps to set up an optimization problem and view results with the Optimization Tool.



3.1.4 Running a Problem in the Optimization Tool

After defining our problem and specifying the options, we are ready to run the solver.



To run the selected solver, click the Start button. For most solvers, as the algorithm runs, the Current iteration field updates. This field does not update for solvers for which the current iteration does not apply.

3.1.5 Pausing and Stopping the Algorithm

While the algorithm is running, we can click Pause to temporarily suspend the algorithm. To resume the algorithm using the current iteration at the time we paused, click Resume.

Click Stop to stop the algorithm. The Run solver and view results window displays information for the current iteration at the moment we clicked Stop. We can export our results after stopping the algorithm.

3.1.6 Viewing Results

When the algorithm terminates, the Run solver and view results window displays the reason the algorithm terminated. To clear the Run solver and view results window between runs, click Clear Results.

3.1.7 Displaying plots

In addition to the Run solver and view results window, we can also display measures of progress while the algorithm executes by generating plots. Each plot selected draws a separate axis in the figure window. We can select a predefined plot function from the Optimization Tool, or we can write our own.

3.1.8 Final Point

The Final point updates to show the coordinates of the final point when the algorithm terminated. If we don't see the final point, then click the upward-pointing triangle on the icon on the lower-left.

3.2 GENETIC ALGORITHM SOLVER

3.2.1 Fitness function (required) is the objective function we want to minimize. We can specify the function as a function handle of the form `@objfun`, where `objfun.m` is an M-file that returns a scalar.

3.2.2 Number of variables (required) is the number of independent variables for the fitness function.

3.2.3 Constraints

Linear inequalities of the form $A*x \leq b$ are specified by the matrix A and the vector b .

Linear equalities of the form $Aeq*x = beq$ are specified by the matrix Aeq and the vector beq .

Bounds are lower and upper bounds on the variables.

Nonlinear constraint function defines the nonlinear constraints. Specify the function as an anonymous function or as a function handle of the form `@nonlcon`, where `nonlcon.m` is an M-file that returns the vectors c and ceq . The nonlinear equalities are of the form $ceq = 0$, and the nonlinear inequalities are of the form $c \leq 0$.

3.2.4 Population

Population options specifies options for the population of the genetic algorithm.

Population type specifies the type of the input to the fitness function. We can set Population type to be Double vector, or Bit string, or Custom. If we select Custom, we must write creation, mutation, and crossover functions that work with population type, and specify these functions in the fields Creation function, Mutation function, and Crossover function, respectively.

Population size specifies how many individuals there are in each generation.

Initial population enables us to specify an initial population for the genetic algorithm.

Initial range specifies lower and upper bounds for the entries of the vectors in the initial population. We can specify Initial range as a matrix with 2 rows and Initial length columns. The first row contains lower bounds for the entries of the vectors in the initial population, while the second row contains upper bounds.

3.2.5 Selection

The selection function chooses parents for the next generation based on their scaled values from the fitness scaling function.

We can specify the function that performs the selection in the Selection function field. We can choose from the following functions:

Stochastic uniform lays out a line in which each parent corresponds to a section of the line of length proportional to its expectation.

Remainder assigns parents deterministically from the integer part of each individual's scaled value and then uses roulette selection on the remaining fractional part.

Roulette simulates a roulette wheel with the area of each segment proportional to its expectation. The algorithm then uses a random number to select one of the sections with a probability equal to its area.

Custom enables we to write our own selection function. Enter a function handle of the form @SelectFcn, where SelectFcn.m is an M-file.

3.2.6 Reproduction

Reproduction options determine how the genetic algorithm creates children at each new generation.

Elite count specifies the number of individuals that are guaranteed to survive to the next generation.

Crossover fraction specifies the fraction of the next generation that crossover produces. Mutation produces the remaining individuals in the next generation. Set Crossover fraction to be a fraction between 0 and 1, either by entering the fraction in the text box, or by moving the slider.

3.2.7 Mutation

Mutation functions make small random changes in the individuals in the population, which provide genetic diversity and enable the genetic algorithm to search a broader space.

3.2.8 Crossover

Crossover combines two individuals, or parents, to form a new individual, or child, for the next generation.

Specify the function that performs the crossover in the Crossover function field. We can choose from the following functions:

Scattered creates a random binary vector. It then selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form the child. For example:

p1 = [a b c d e f g h]

p2 = [1 2 3 4 5 6 7 8]

random crossover vector =

[1 1 0 0 1 0 0 0]

child = [a b 3 4 e 6 7 8]

Single point chooses a random integer n between 1 and Number of variables, and selects the vector entries numbered less than or equal to n from the first parent, selects genes numbered greater than n from the second parent, and concatenates these entries to form the child. For example:

p1 = [a b c d e f g h]

p2 = [1 2 3 4 5 6 7 8]

random crossover point = 3

child = [a b c 4 5 6 7 8]

Custom enables us to write our own crossover function that satisfies any constraints specified. Enter a function handle of the form @CrossoverFcn, where CrossoverFcn.m is an M-file.

3.2.9 Hybrid function

Hybrid function enables us to specify another minimization function that runs after the genetic algorithm terminates. The choices available are:

None

fminsearch (unconstrained only)

patternsearch (constrained or unconstrained)

fminunc (unconstrained only)

fmincon (constrained only)

3.2.10 Stopping criteria

Stopping criteria determines what causes the algorithm to terminate.

Generations specifies the maximum number of iterations the genetic algorithm performs.

Time limit specifies the maximum time in seconds the genetic algorithm runs before stopping.

Fitness limit — If the best fitness value is less than or equal to the value of Fitness limit, the algorithm stops.

Stall generations — If the weighted average change in the fitness function value over Stall generations is less than Function tolerance, the algorithm stops.

Stall time limit — If there is no improvement in the best fitness value for an interval of time in seconds specified by Stall time limit, the algorithm stops.

Function tolerance — If the cumulative change in the fitness function value over Stall generations is less than Function tolerance, the algorithm stops.

Nonlinear constraint tolerance specifies the termination tolerance for the maximum nonlinear constraint violation.

3.2.11 Plot Functions

Plot functions enable we to plot various aspects of the genetic algorithm as it is executing. Each one draws in a separate axis on the display window. Use the Stop button on the window to interrupt a running process.

Plot interval specifies the number of generations between successive updates of the plot.

Best fitness plots the best function value in each generation versus iteration number.

Best individual plots the vector entries of the individual with the best fitness function value in each generation.

Distance plots the average distance between individuals at each generation.

Expectation plots the expected number of children versus the raw scores at each generation.

Scores plots the scores of the individuals at each generation.

Selection plots a histogram of the parents. This shows us which parents are contributing to each generation.

Custom Enter a function handle of the form @plotfun, where plotfun.m is an M-file.

3.3 SOLVING ROSENBROCK FUNCTION USING OPTIMTOOL-BOX

3.3.1 Rosenbrock Function

Minimize $F(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$

Range $x_1 \in (0, 2)$

$x_2 \in (0, 2)$

3.3.2 Algorithm to Solve Rosenbrock Function by Ga Toolbox

1. Write function in M. FILE in proper syntax
[fun f= name(x)
f=();
a=x(1);
a=x(2);
z=[a,b];
Disp(z)]
2. Open optimtool
Limitation:-product of population size & no. of iterations ≤ 2500
Fill values in optimtool
This give values in command window.
3. Copy all values to new M.FILE & make matrix according to the desired iteration value.
Let it be 40th iteration & name it g40
4. Transfer g40 to command window & multiply it by matrix $\text{transp}[1\ 0]$ & $\text{transp}[0\ 1]$ for x40 & y40
5. Plot[x(1)40,x(2)40,'d']

where "d" is style option

3.3.4 M-file for objective function

```
function f = twovar(x)
f = 100 * ((x(1)^2-x(2))^2)+((1-x(1))^2);
a=x(1);
b=x(2);
z=[a,b];
disp(z)
```

3.3.5 Optimtool Specifications For Minimization Of Rosenbrock Function Problem

After opening the OPTIMTOOL-BOX, the various values of input parameters in OPTIMTOOL-BOX are following:-

Solver- ga Genetic Algorithm

Constraints columns are left blank since Rosenbrock function solved for this problem has no constraints.

Algorithm setting and all Population options like Population type, Population size, Initial population, Initial scores ,Initial range are set at default values

Scaling function-Rank

Selection Function-Stochastic Uniform

Hybrid function-fmincon

Stopping Criteria

Generation-350

Stall generation-100

Function tolerance- 1×10^{-5} and rest values are taken as default.

3.4 RESULTS

DIFFERENT POINTS OBTAINED AFTER EVERY 40TH ITERATION USING GA TOOLBOX.		
g1 (1 st iteration)	1.7078 1.0930 1.2777 0.6319 0.1558 0.0234 1.2745	0.9623 0.0928 1.1010 0.6856 0.0404 1.2892 1.0992
g40 (40 th iteration)	1.0192 1.0192 1.0192 1.0192 0.7452 1.0192 1.6713	0.9555 0.9555 0.9555 0.9555 -0.5548 0.9555 1.5474
g80 (80 th iteration)	1.5974 1.5974 1.0620 1.5974 1.5974 1.5974 3.6916	2.5020 2.5020 1.6995 2.5020 2.1692 2.5020 1.4259
g120 (120 th iteration)	1.5974	2.5020

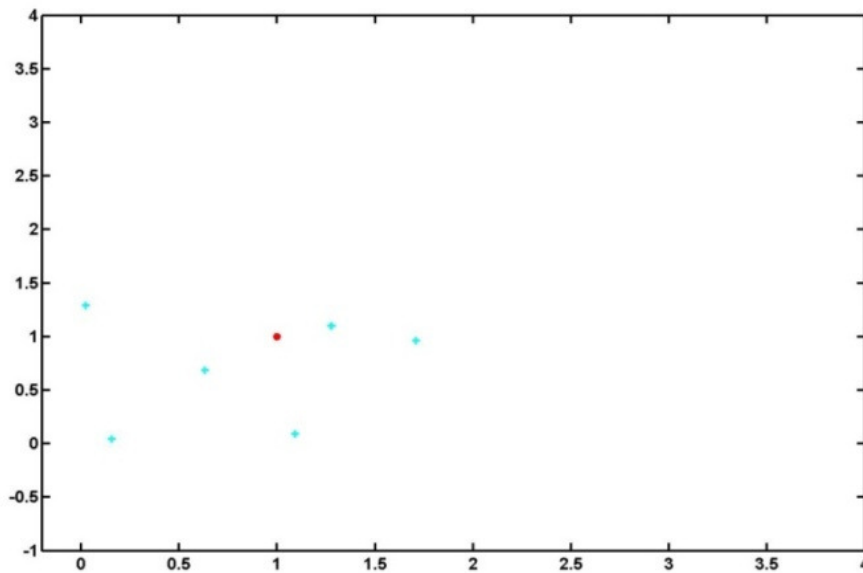
GA BASED MULTIOBJECTIVE ECONOMIC LOAD DISPATCH

	1.5974	2.5020
	1.5974	2.5020
	1.5974	2.5020
	1.5974	2.5020
	1.1930	2.5020
	3.6703	3.8612
g160 (160 th iteration)	1.5752	2.5020
	1.5752	2.5020
	2.5174	2.5020
	1.5752	2.5020
	1.5752	2.5020
	2.0180	1.3965
	1.5752	2.5020
g200 (200 th iteration)	1.5752	2.5020
	1.5752	2.5020
	1.1107	2.4558
	1.5752	2.5020
	-0.0116	2.5020
	1.5752	2.5020
	1.5329	1.6214
g240 (240 th iteration)	1.0694	1.1809
	1.0694	1.1809
	1.0694	1.1809
	1.0694	1.1809
	1.0694	1.1809
	1.0694	1.1809
	2.1159	0.8682
g280 (280 th iteration)	0.9532	0.8994
	0.9532	0.8994
	0.9532	0.8994
	0.9532	0.8994
	0.9532	0.8994
	0.9532	0.9755
	0.5783	0.9345
g320 (320 th iteration)	0.9532	0.9054
	0.9532	0.9054
	0.9532	0.9832
	0.8893	0.9054
	0.9532	0.9054
	0.9532	0.9054
	0.8451	0.9086
g350 (350 th iteration)	0.9572	0.9152
	0.9532	0.9054
	0.9572	0.9054
	0.9532	0.9054
	0.9751	0.9054
	0.9532	0.9054
	0.9532	0.9054

3.5 PLOTS

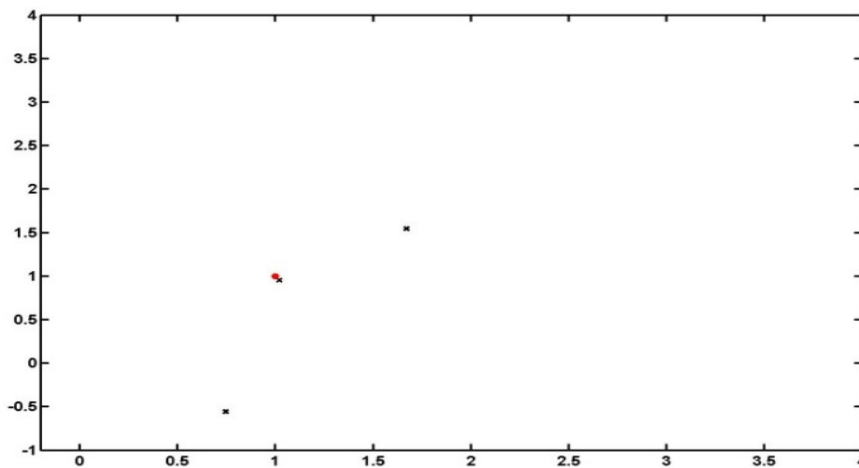
Following are the Plots drawn for every 40th iteration mentioned in the table shown in preceding section(3.4).Here Red colour dot indicate the ideal point i.e.(1,1) and blue colour dot indicate the points obtained by running problem in OPTIMTOOL-BOX.

(i) 1st iteration:-

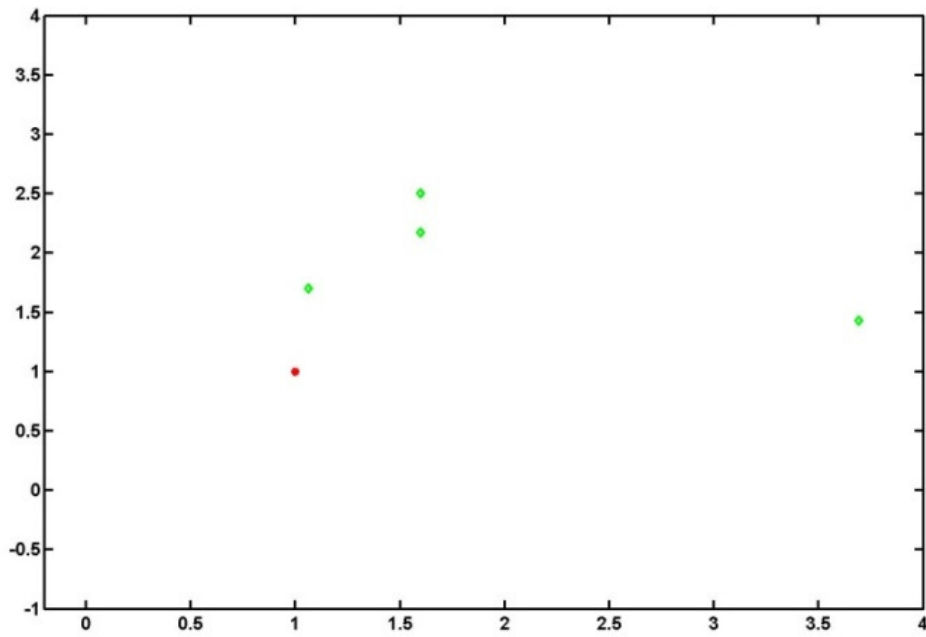


Remark on 1st iteration plot :Points obtained are scattered around the optimum point(i.e [1,1])

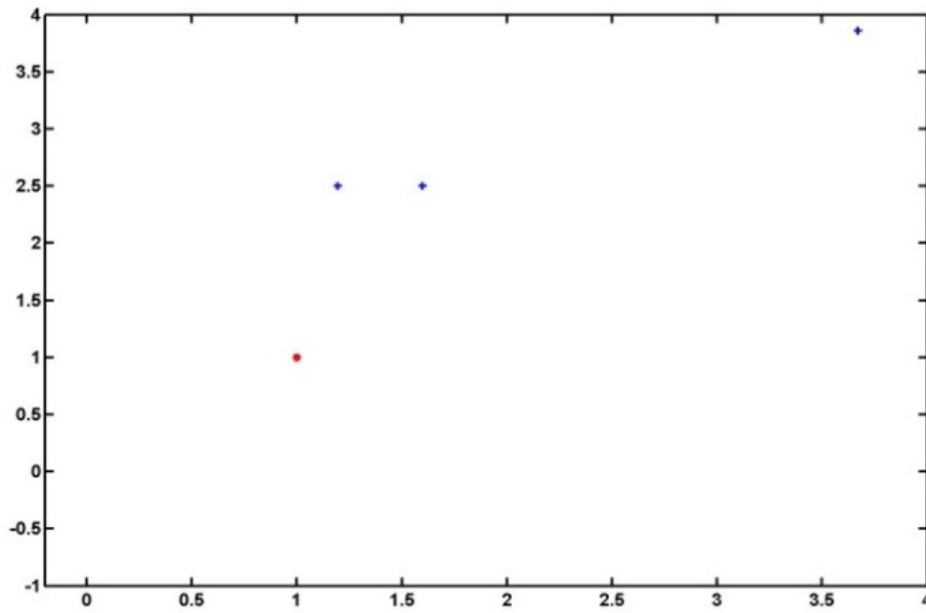
(ii) 40th iteration:-



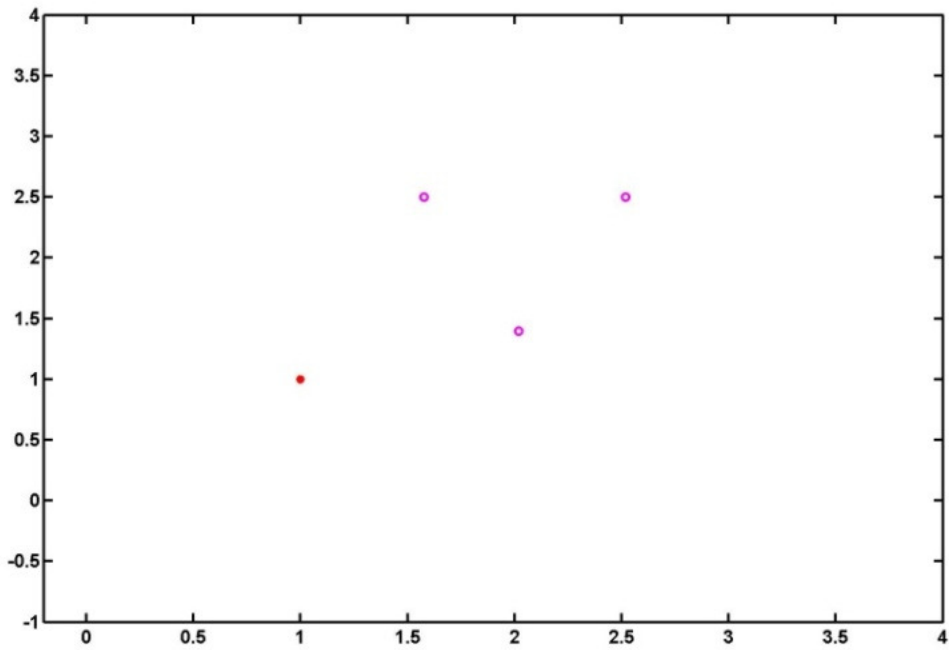
(iii) 80th iteration:-



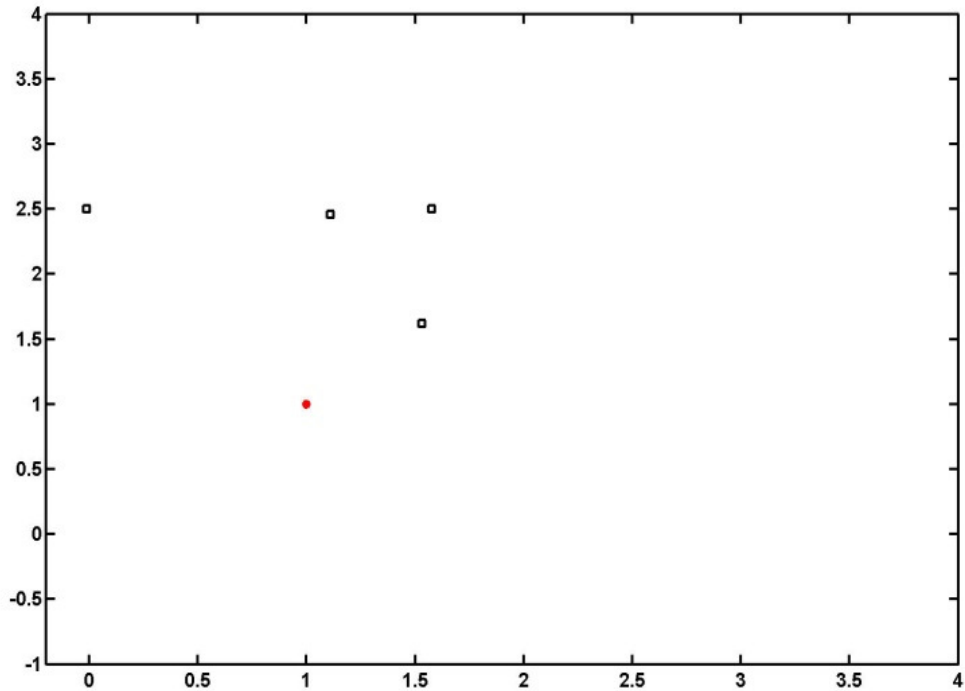
(iv) 120th iteration:-



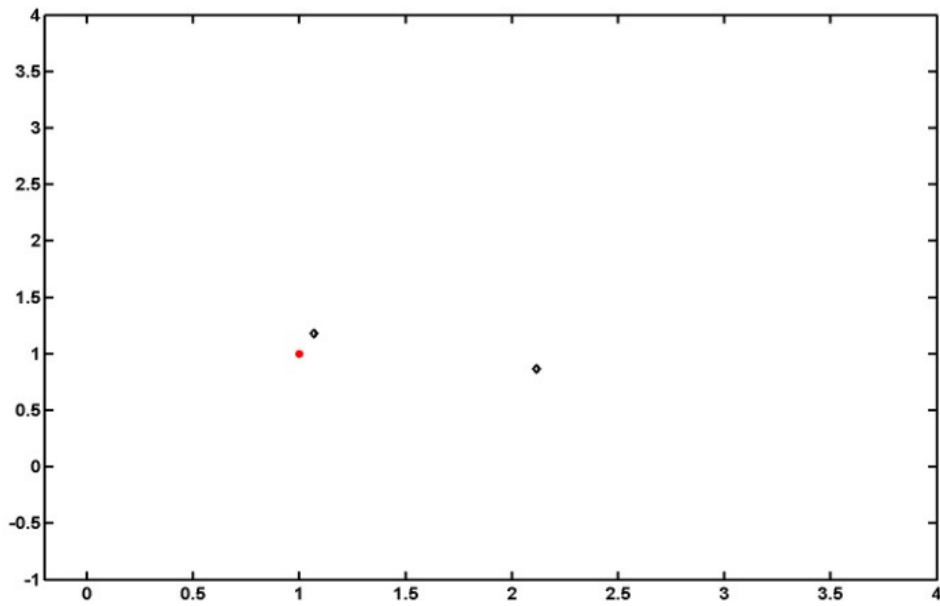
(v) 160th iteration:-



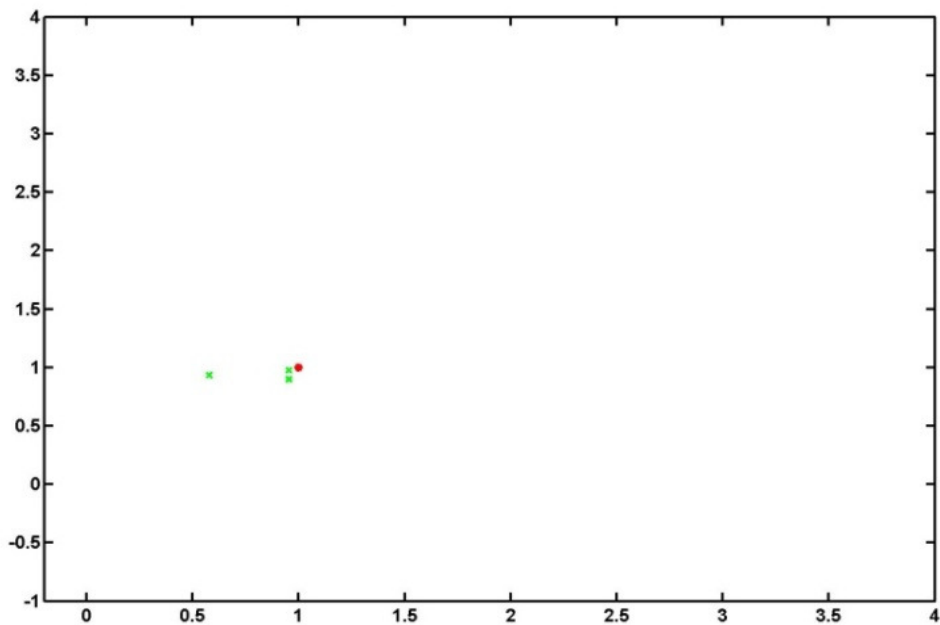
(vi) 200th iteration:-



(vii) 240th iteration:-

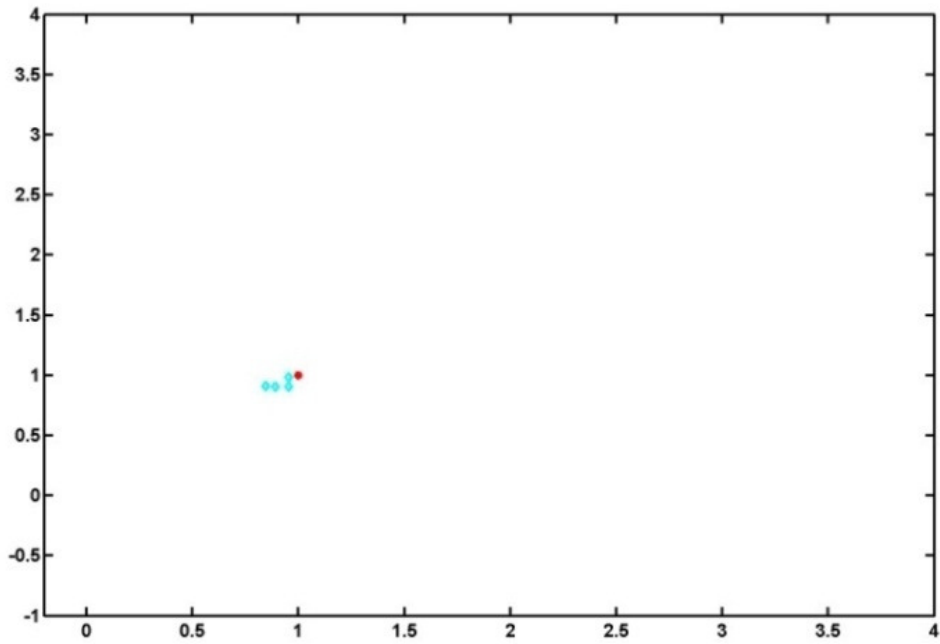


(viii) 280th iteration:-

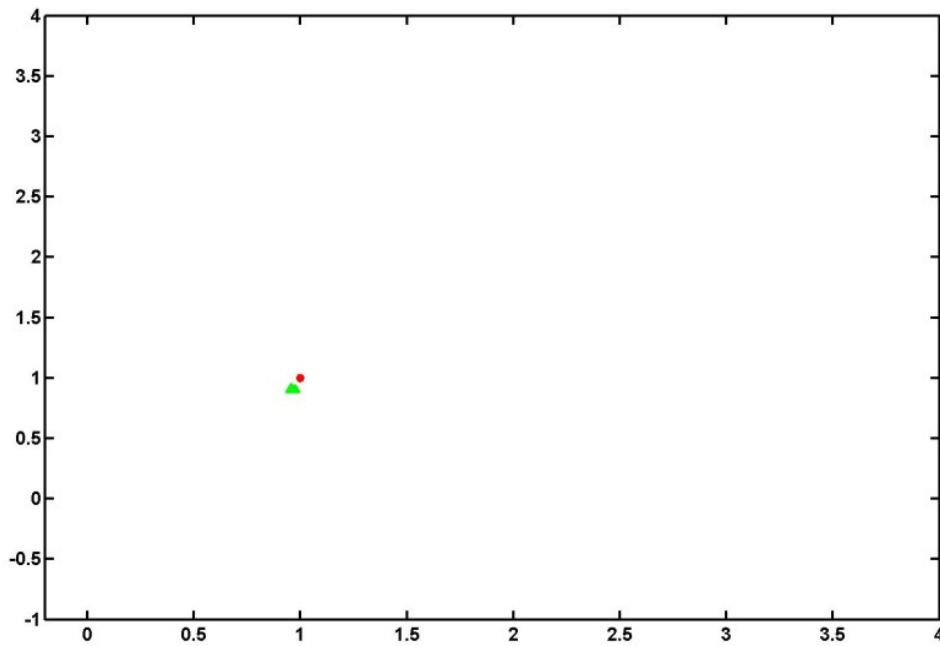


Remark on 280th iteration plot: Points are converging to the optimum point(1,1)

(ix) 320th iteration:-



(x) 350th iteration:



3.6 CONCLUSION

Following are the significant contribution of the research work presented in this chapter:-

While solving Rosenbrock function using GA toolbox, as we know the optimum point of function happens to be (1,1) {x1,x2}. In 40th iteration, we have a point in population as (1,1) but at this stage it can't be identified as optimum point.

As generation count progress, it is observed that more & more number of population comes close in vicinity of (1,1).

In the last generation i.e. 350th, it is observed that members of population are in close proximity of (1,1) within function tolerance value $1*10^{-5}$.

CHAPTER 4

MULTIOBJECTIVE OPTIMIZATION

4.1 INTRODUCTION

The various objectives of power systems are cost of generation, system transmission losses, environmental pollution, security etc. These objectives are conflicting in nature and cannot be handled by conventional single objective optimization techniques. Single objective technique gives optimal solution in respect of an objective function under consideration. The way out, therefore, lies in the multiobjective approach to problem solving.

4.2 FORMULATION OF GENERAL MULTIOBJECTIVE PROGRAMMING PROBLEM:

The general multiobjective optimization problem with n decision variables, m constraints and h objectives is

Minimize

$$Z(X_1, X_2, \dots, X_n) = [Z_1(X_1, X_2, \dots, X_n); \quad (4.1a)$$

$$Z_2(X_1, X_2, \dots, X_n);$$

.....;

$$Z_h(X_1, X_2, \dots, X_n)];$$

s.t.

$$g_i(X_1, X_2, \dots, X_n) \leq 0 \quad i = 1, 2, \dots, m \quad (4.1b)$$

$$X_j \geq 0 \quad j = 1, 2, \dots, n \quad (4.1c)$$

Where $Z(X_1, X_2, \dots, X_n)$ is the multiobjective function and $Z_1(X_1, X_2, \dots, X_n), Z_2(X_1, X_2, \dots, X_n), \dots, Z_h(X_1, X_2, \dots, X_n)$ are the h individual objective functions. In the multiobjective function Z , the various individual functions Z_1, Z_2, \dots, Z_h have just been written, but it does not imply any kind of operation say multiplication, addition or anything else whatsoever in general. In particular, Z can be designed to incorporate Z_1, Z_2, \dots, Z_h depending upon the approach.

Multiobjective approach to economic load dispatch has been carried out on IEEE 5 bus system in 3D space. The data of IEEE 5 bus system is given in Appendix I. In 3D space, three objectives i.e. cost of generation (F_C), system transmission losses (F_L) and environmental pollution (F_P) have been considered. The ideal situation where one would like to operate the power systems is one where all the objectives i.e. cost of generation (F_C), system transmission losses (F_L) and environmental pollution (F_P) are minimum. Such a point is called the ***Ideal Point***. In 3D space, it is represented by $(F_{Cmin}, F_{Lmin}, F_{Pmin})$. But such a point is not feasible. If it was, then there would not be any conflict among the objectives.

A strategy has to be adopted by the power system operator to achieve optimum values as per his satisfaction level and requirements. The operating point so obtained is called ***Target Point*** (TP) or the best-compromise solution.

4.3 NONINFERIORITY

A feasible solution to a multiobjective programming problem is noninferior if there exist no other feasible solution that will yield an improvement in one objective without causing degradation in at least one of the other objectives [3]. A given noninferior solution may or may not be acceptable to the decision maker. However, it is important to note that, it is one of these noninferior solutions for which decision maker looks for.

4.3.1 Graphical Explanation of Noninferiority

Let us explain this definition graphically. An arbitrary collection of feasible alternatives for a two objective minimization problem is shown in Fig 4.1. Curve 1 forms the boundary of the feasible region. The definition of noninferiority can be used to find noninferior solutions in Fig 4.1. All the feasible solutions above curve 1 are inferior because they yield more of both $Z_1 (F_C)$ and $Z_2 (F_L)$. Consider an exterior point C in Fig 4.1, which is inferior. Alternative A gives less of $Z_1 (F_C)$ than does C without increasing the amount of $Z_2 (F_L)$. Alternative B gives less amount of $Z_2 (F_L)$ without increasing the amount of $Z_1 (F_C)$. Consider point D on curve AB. Suppose it is desired to achieve lesser value of $Z_1 (F_C)$ than the value at point D. Since it is not desirable to move to the left of curve AB as even though it gives lesser value of $Z_1 (F_C)$, yet it lies in the infeasible region. Therefore, it is desirable to move upward only along the curve AB to have lesser value of $Z_1 (F_C)$. Let us say, we get point E. At this point, we get less value of $Z_1 (F_C)$ but there is some increase in $Z_2 (F_L)$. In other words, in order to gain on $Z_1 (F_C)$, we have to sacrifice $\Delta Z_2 (\Delta F_L)$ units of $Z_2 (F_L)$. Similarly, in moving from D to F, we have to sacrifice $\Delta Z_1 (\Delta F_C)$ units of $Z_1 (F_C)$ to gain on $Z_2 (F_L)$. So we can say that points D, E and F are noninferior.

4.3.2 Mathematical Definition of Noninferiority:-

Single objective problems are characterized by complete ordering of their feasible solutions. Any two feasible solutions X_1 and X_2 are comparable in terms of the objective function; i.e. either

$$Z(X^1) = Z(X^2), Z(X^1) > Z(X^2), Z(X^1) < Z(X^2).$$

This comparison can be made for all the feasible solutions, and the solution X^* for which there exists no other solution X such that $Z(X) < Z(X^*)$ is called optimal solution for a minimization problem. But, in multiobjective problems, it is not possible to compare all the feasible solutions because the comparison on the basis of one objective function may contradict the comparison based on another objective function.

Suppose there are two objective functions,

$$Z(X) = [Z_1(X), Z_2(X)]$$

and two solutions X^1, X^2 . Then,

$$Z(X^1) = [Z_1(X^1), Z_2(X^1)]$$

$$Z(X^2) = [Z_1(X^2), Z_2(X^2)]$$

X^1 is better than X^2 if

$$Z_1(X^1) < Z_1(X^2) \quad \text{and} \quad Z_2(X^1) \leq Z_2(X^2)$$

or

$$Z_1(X^1) \leq Z_1(X^2) \quad \text{and} \quad Z_2(X^1) < Z_2(X^2)$$

but if $Z_1(X^1) < Z_1(X^2)$ AND $Z_2(X^1) > Z_2(X^2)$, then nothing can be said about the two solutions – X^1 , X^2 , i.e. they are incomparable. This is what is meant by partial ordering. All solutions are not comparable on the basis of the values objective functions only. Since a complete order is not available, the notion of optimality must be dropped.

The partial ordering in multiobjective problems does not allow some feasible solutions to be eliminated. Inferior solutions, which are dominated by at least one feasible solution, may be dropped. Noninferior solutions are the alternatives of interest.

Mathematically, a solution X is noninferior for a minimization problem if there exist no feasible Y such that

$$Z_K(Y) \leq Z_K(X) \quad \forall K = 1, 2, \dots, H$$

and

$$Z_K(Y) < Z_K(X) \quad \text{for at least one } K = 1, 2, \dots, h$$

The noninferior set generally includes many alternatives, all of which obviously cannot be selected. The objectives must be traded off against each other in moving from one noninferior alternative to another and a strategy has to be adopted by the analyzer to achieve optimum values as per his satisfaction level and requirements. The preferred alternative is called **Target Point** or the best compromise solution.

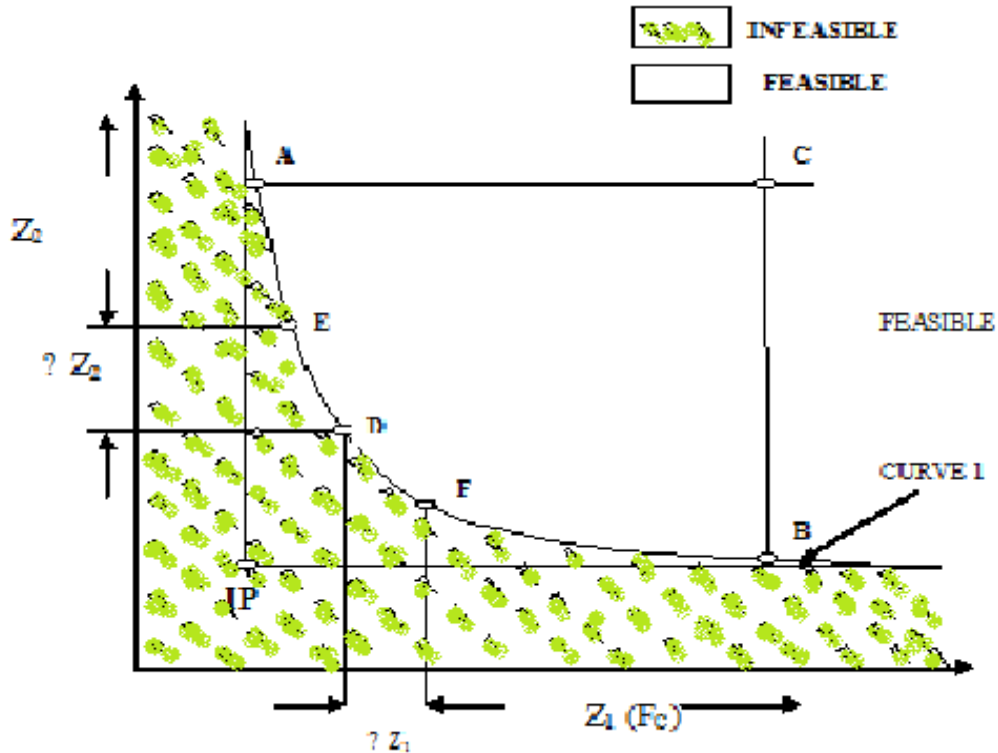


Fig 4.1: FEASIBLE REGION IN OBJECTIVE SPACE

4.4 WEIGHTING METHOD:-

Weighting the objective to obtain noninferior solution is the oldest multiobjective solution technique [3]. The method follows directly from the necessary conditions of Noninferiority developed by Kuhn and Tucker [3]. Gass and Saaty [3] showed how noninferior solutions could be generated in two-objective problems by parametrically varying the objective function coefficients. Zadeh [3] was the first to recommend the use of weights to approximate the noninferior set. Marglin and Major [3] discussed the use of weighting in multiobjective public investment problems.

Suppose, for example that we have a fire station location problem for which there are two objectives: maximize the property value (measured in dollars) within S miles of the facility and maximize the population within S miles of the facility. The property value and population objective will be called Z_1 and Z_2 , respectively. The two objectives conflicts because commercial

areas are characterized by high property value and low populations while residential areas have more people and lower property value. Since the fire station cannot be located such that the entire area is within S miles, the maxima of Z_1 and Z_2 cannot be obtained simultaneously. The objective function for this multiobjective location problem is

Maximize $Z=[Z_1,Z_2]$

Where, Z_1 =property value

Z_2 =population objective

Now, for the comparison of the value judgment between population and property value, one person is assigned with value worth ' w ' dollars. Then the multiobjective problem could be reduced to a single-objective problem. The specifications of w , which is called weight on objective Z_2 (population), is equivalent to the identification of a desirable tradeoff between Z_1 and Z_2 . Since we know the value of Z_2 in terms of Z_1 the equation can be rewritten as

Maximize $Z(w)=Z_1+wZ_2$

Now the objective function has a single dimension and is denoted by $Z(w)$ to signify the dependence of the new function on the values of the weight w . The units of the new objective function $Z(w)$ are dollars: Z_1 is measured in dollars and wZ_2 is (dollars/persons)*(persons) = dollars. Now, depending on the value of ' w ', desirable tradeoff between Z_1 and Z_2 can be achieved.

4.5 IDEAL DISTANCE MINIMIZATION METHOD

This method [5] employs the concept of an 'Ideal Point'(IP) to scalarize the problem having multiple objective and minimizes the Euclidean distance between the IP and set of feasible or noninferior solution.

The ideal solution where one would like to operate the power system is the one where all the three objectives namely cost of generation(F_C), system transmission loss (F_L) and pollution(F_P) are minimum. In a 3D space where three axis represent three objective functions, i.e having the coordinates as $(F_{Cmin}, F_{Lmin}, F_{Pmin})$ is known as IP, which is not feasible. Therefore, one can at most achieve a point which is feasible and at a minimum distance from the IP. Such a point is named as Target Point (TP) or the best compromise solution. In order to locate this TP the following distance function for MOELD problem in 3D space is proposed:

$$\text{Distance} = [(F_C - F_{Cmin})^2 + (F_L - F_{Lmin})^2 + (F_P - F_{Pmin})^2]^{1/2} \quad \dots 4.2$$

Where,

F_{Cmin} is the minimum value of cost of generation in 3D space,

F_{Lmin} the minimum value of system transmission loss in 3D space and

F_{Pmin} is the minimum value of pollution in 3D space.

CHAPTER 5

MULTI-OBJECTIVE ECONOMIC LOAD DISPATCH IN 3D SPACE

5.1 INTRODUCTION

Electrical energy cannot be stored but is generated from natural sources and delivered as the demand raises. A transmission system is used for the delivery of bulk power over considerable distances. The power system consists of three parts, generator, which produces electricity, transmission line, which transmits it to far-away places and load, which uses it. This configuration is applicable to all the interconnected networks but the number of elements may vary. The transmission networks are interconnected through tie lines so that utilities may interchange power, share reserve and render assistance to one another at the time of need. Since the sources of energy are so diverse, so the choice of the required sources is made on economic, technical and geographical basis. As there are few facilities to store electrical energy, the net production of a utility must clearly track its total load. For an interconnected system, it is necessary to minimize the expenses. The economic load dispatch (ELD) is used to define the production level of each plant, so that the total cost of generation and transmission is minimum for a prescribed schedule of load or ELD may also be defined as the process of allocating generation levels to the generating units in the mix, so that the system load may be supplied entirely and most economically.

5.2 LOAD DISPATCHING

Nowadays operation of a modern power system has become very complex. It is necessary to maintain frequency and voltage within limits, which is done by matching the generation of active and reactive power with the load demand. In addition, for ensuring reliability of power system it is mandatory to put additional generation capacity into the system in the event of outage of generating equipment at some station. Above all cost of electric supply should be ensured at minimum. The total interconnected network is controlled by the load dispatch centre which allocates the MW generation to each grid depending upon the prevailing MW demand in that area. Each load dispatch center controls load and frequency of its own by matching generation in various generating stations with total required MW demand plus MW losses. Therefore, the task of load control center is to keep the exchange of power between various zones and system frequency at desired values.

5.3 ECONOMICS OF POWER GENERATION

In all engineering works, the question of cost is of first importance. The electrical power supplier is required to supply power to a large number of consumers to meet their requirements. While designing electrical power generating stations and other systems efforts are made to achieve overall economy so that per unit cost of generation is the lowest possible. This will enable the supplier to supply electrical energy to its consumer at reasonable rates. The cost depends on the number of hours the plant is in operation or upon the number of units of electrical energy generated i.e. the operating cost is approximately proportional to units generated. Total annual cost incurred in the power generation is represented by the expression (5.1).

$$F_C = F[C_i (P_{gi})] = \sum_{i=1}^{N_g} (a_i P_{gi}^2 + b_i P_{gi} + c_i) \quad (5.1)$$

Where, $i=1,2,\dots,N_g$

N_g =number of generators

a_i, b_i, c_i are the Cost coefficients of the i^{th} generating unit(see Appendix).

The factors influencing power generation at minimum cost are operating efficiencies of generators, fuel cost and transmission losses. The most efficient generator in the system does not guarantee minimum cost as it may be located in an area where fuel cost is high. Also, if the plant is located far from the load centre, transmission losses may be considerably higher and hence, the plant may be overly uneconomical. Hence, the problem is to determine the generation of different plants such that the total operating cost is minimum. The operating cost plays an important role in the economic scheduling.

The cost of fuel used for economics of power generation is specified by the input-output curve of a generating unit. The input to the thermal plant is generally measured in BTU/hr and the output is measured in MW. A simplified input output curve of the thermal unit known as heat rate curve is given in following Fig. 5.1(a). Converting the ordinate of heat rate curve from BTU/hr to Rs/hr. results in the fuel cost curve shown in Fig. 5.1(b)

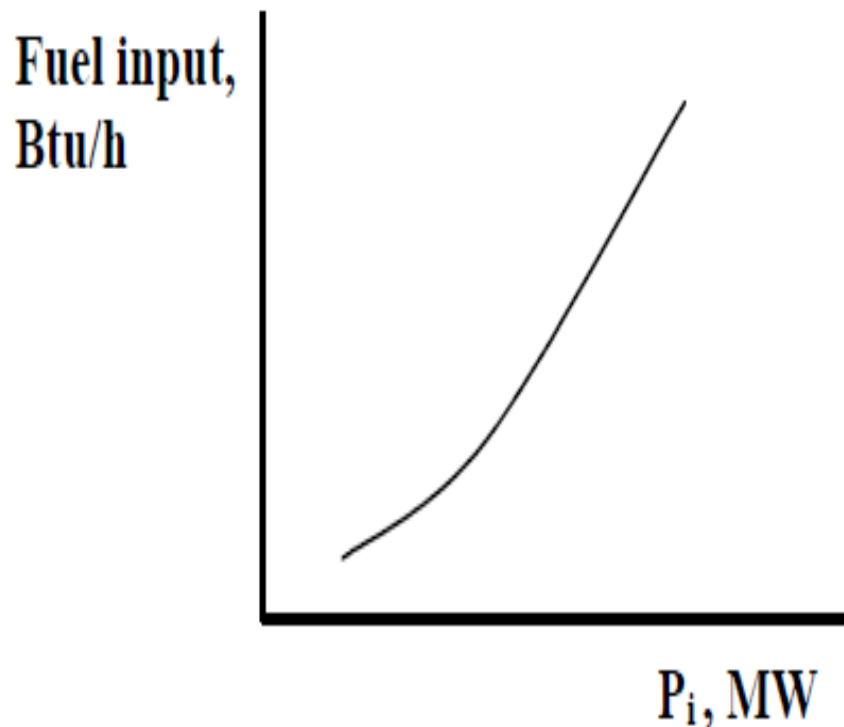


Fig. 5.1(a) Heat-rate curve



Fig 5.1(b) Fuel-rate curve

In all practical cases, the fuel cost of generator i can be represented as a quadratic function of real power generation from equation 5.1. An important characteristic is obtained by plotting the derivative of fuel cost curve vs. real power. This is known as the incremental fuel cost curve shown in Fig. 5.1(c).

$$dC_i/dP_i = 2a_i P_i + b_i$$

The incremental fuel cost curve is a measure of how costly it will be to produce the next increment of power. The total operating cost includes the fuel cost, and the cost of labor, supplies and maintenance. These costs are assumed to be a fixed percentage of the fuel cost and are generally included in the incremental fuel cost curve.

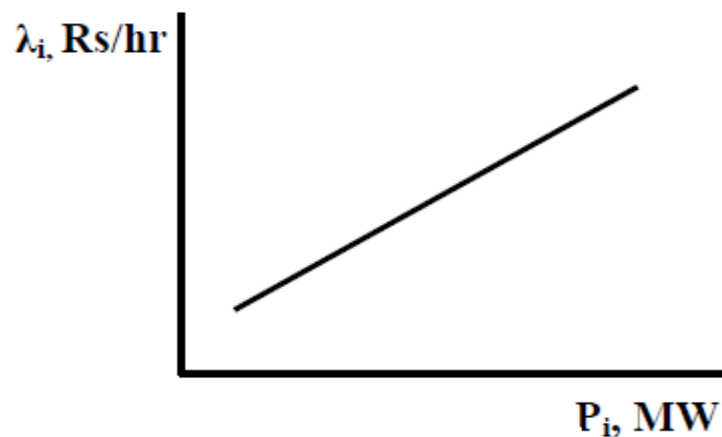


Fig. 5.1(c) Incremental fuel-cost curve

5.4 TRANSMISSION LOSSES

When transmission distances are very small and load density is very high, transmission losses may be neglected and the optimal dispatch of generation is achieved with all plants operating at equal incremental production cost. However, in a large inter connected network where power is transmitted over long distances with low load density areas, transmission losses are major factor and affect the optimum dispatch of generation. One common practice for including the effect of transmission losses is to express the total transmission loss as a quadratic function of the generator power outputs. The simplest quadratic form is

$$F_L = \sum \sum P_i B_{ij} P_j$$

$$(i, j=1, 2, \dots, N_g)$$

Where $i, j =$ number of generating units or plants i.e. $i, j=1, 2, 3, \dots, N_g$

Where $N_g =$ number of generators.

A more general formula containing a linear term and a constant term, referred to the *Kron's loss formula*, is

$$F_L = \sum \sum P_i B_{ij} P_j + \sum B_{0i} P_i + B_{00}$$

The B terms are called loss coefficients or B-coefficients and for N bus system, $N \times N$ square matrix B which is always symmetrical, is known as the B-matrix. The unit of the B_{ij} is reciprocal megawatts when the three-phase power P_i and P_j are expressed in megawatts, in which case F_L will be in megawatts also. The units of B_{00} match those of F_L while B_{i0} is dimensionless. In this work B_{00} is assumed negligible.

These B coefficients for a given system are assumed to remain constant, and reasonable accuracy can be expected provided the actual operating conditions are close to the base case where the B-constants are computed [51].

5.4.1 Formulation of Economic Load Dispatch Problem

Mathematically, the Economic Load Dispatch Problem is expressed as:

$$\text{Minimize, } F_C = \sum_{i=1}^{NG} [F\{C_i (P_{gi})\}] \quad (i=1, 2, \dots, N_g) \quad (5.1)$$

$$\text{s.t.} \quad \sum_{i=1}^{NG} [P_{gi}] = P_D + F_L \quad (i=1, 2, \dots, N_g) \quad (5.2)$$

$$P_{gimin} \leq P_{gi} \leq P_{gimax} \quad (5.3)$$

Where P_g is the power generation at i^{th} generator

P_D is power demand

F_L is function representing the system transmission loss.

5.5 ENVIRONMENTAL POLLUTION

The combustion of fuel used in fossil based generating units, gives rise to three basic forms of pollutants [52]. These are: oxides of sulphur (SO_x), oxides of nitrogen (NO_x), and carbon dioxide (CO_2), which are very harmful for human as well as other life forms. Therefore, it is necessary to reduce pollution level. The US Clean Air Act Amendments (CAAA) of 1990 [53] has made mandatory for the Electric utilities to significantly reduce pollution levels from those of 1980 levels. The overall goal for the NO_x emission is to lower NO_x emission by 2 million tons per year. In the present work, oxides of nitrogen emission is taken as the index for environmental pollution. It is given as a function of generator output

$$F_p = \sum_{i=1}^{NG} (d_i P_i^2 + e_i P_i + f_i)$$

Where, $i=1,2,\dots,N_g$

N_g =number of generators and

d_i, e_i, f_i are the pollution coefficients of the i^{th} generating unit (see Appendix).

5.6 FORMULATION OF MOELD PROBLEM IN 3D SPACE

Three aspects of the Multiobjective Economic Load Dispatch (MOELD) problem considered in 3D space are:

- 1- To minimize the cost of generation.
- 2- To minimize the system transmission losses.
- 3- To minimize the environmental pollution.

The objective function to minimize the cost of generation is given as,

$$F_C = \sum F[C_i(P_{gi})] \quad (i=1,2,\dots,N_g) \quad (5.4)$$

Where, P_{gi} is the power generation at the i^{th} generator, C_i is the cost of generation for i^{th} generator and N_g is the total number of generators in the system.

The objective function to minimize system transmission loss is given as

$$F_L = \sum \sum P_i B_{ij} P_j \quad (5.5)$$

where $i, j=1,2,3,\dots,N_g$

N_g = number of generators.

The objective function to minimize environmental pollution is given as

$$F_P = \sum_{i=1}^{NG} (d_i P_i^2 + e_i P_i + f_i) \quad (5.6)$$

Where, $i=1,2,\dots,N_g$

N_g = number of generators

In 3D space, the multiobjective function F comprises of cost of generation, system transmission losses and environmental pollution i.e.

$$F = [F_C, F_L, F_P]$$

To generate the noninferior solution of multiobjective optimization problem, the weighting method is used. In this method the problem is converted into a scalar optimization problem as

Minimize

$$F = W_C F_C + W_L F_L + W_P F_P \quad (5.7)$$

s.t equality and inequality constraints as defined by eqns (5.2) & (5.3)

where, F_C is the cost of generation

W_C is the Weight attached to the cost of generation

F_L is the System transmission loss

W_L is the weight attached to the system transmission losses

F_P is the environmental pollution

W_P is the Weight attached to the environmental pollution.

CHAPTER 6**RESULTS & DISCUSSION****6.1 INTRODUCTION**

The Multiobjective Economic Load Dispatch (MOELD) is formulated in 3D space by eq(5.7). The noninferior set is generated by keeping W_c fixed to 1.0(one) and varying weights attached to transmission loss (W_L) and environmental pollution (W_p). Table 6.1 shows the noninferior set of IEEE 5 bus system in 3D space.

TABLE 6.1

NONINFERIOR SET OF IEEE 5-BUS SYSTEM IN 3D SPACE

S.no	Generation cost		Transmission Loss		Environmental Pollution		Distance
	W_c	F_c	W_L	F_L	W_p	F_p	
1	1	760.6602	0	5.1811	0	599.6354	1.6525
2	1	760.6805	0	5.2058	0.1	599.1924	1.2141
3	1	760.7223	0	5.2259	0.2	598.906	0.9357
4	1	760.8643	0	5.2674	0.5	598.4691	0.5633
5	1	761.0441	0	5.3052	1	598.213	0.5092
6	1	761.4349	0	5.3702	5	598.0056	0.8354
7	1	761.4996	0	5.3798	8	597.9952	0.8989
8	1	761.0445	0	5.3053	10	598.2125	0.5093
9	1	761.5728	0	5.3903	20	597.9889	0.9711
10	1	761.6158	0	5.3964	50	597.9876	1.0137
11	1	761.6225	0	5.3974	100	597.9875	1.0203
12	1	761.6248	0	5.3977	250	597.9875	1.0226
13	1	760.6618	0	5.1744	500	599.7745	1.7908
14	1	760.6662	0.5	5.1943	0	599.3853	1.4044
15	1	760.6662	0.5	5.1943	0.1	599.3853	1.4044
16	1	760.7059	0.5	5.2191	0.2	598.9966	1.0229
17	1	760.8398	0.5	5.2613	0.5	598.5212	0.5986
18	1	761.0179	0.5	5.3002	1	598.2405	0.5005
19	1	761.4225	0.5	5.3683	5	598.0082	0.8232
20	1	761.4911	0.5	5.3785	8	597.9963	0.8905
21	1	761.5161	0.5	5.3822	10	597.9933	0.9152

GA BASED MULTIOBJECTIVE ECONOMIC LOAD DISPATCH

22	1	761.5691	0.5	5.3898	20	597.989	0.9675
23	1	761.6017	0.5	5.3944	50	597.9878	0.9997
24	1	761.615	0.5	5.3963	100	597.9876	1.0129
25	1	761.6222	0.5	5.3973	250	597.9876	1.02
26	1	761.6246	0.5	5.3976	500	597.9875	1.0224
27	1	760.6664	1	5.1684	0	599.91	1.9257
28	1	760.6649	1	5.1927	0.1	599.4131	1.4319
29	1	760.6928	1	5.2129	0.2	599.0857	1.1095
30	1	760.8392	1	5.2612	0.5	598.5226	0.5997
31	1	760.9936	1	5.2954	1	598.2684	0.4963
32	1	761.0184	1	5.3003	5	598.24	0.5007
33	1	761.4828	1	5.3773	8	597.9974	0.8824
34	1	761.5092	1	5.3812	10	597.994	0.9084
35	1	761.5365	1	5.3851	20	597.9913	0.9353
36	1	761.6017	1	5.3944	50	597.9878	0.9997
37	1	761.608	1	5.3953	100	597.9877	1.006
38	1	761.6194	1	5.3969	250	597.9876	1.0172
39	1	761.6232	1	5.3974	500	597.9875	1.021
40	1	760.765	5	5.1333	0	600.8982	2.9136
41	1	760.6648	5	5.1926	0.1	599.4155	1.4343
42	1	760.6928	5	5.2129	0.2	599.0857	1.1095
43	1	760.6995	5	5.2161	0.5	599.0381	1.0631
44	1	760.9929	5	5.2952	1	598.2692	0.4962
45	1	761.3208	5	5.3526	5	598.037	0.7249
46	1	761.4194	5	5.3679	8	598.0088	0.8202
47	1	761.4561	5	5.3733	10	598.0016	0.8561
48	1	761.5365	5	5.3851	20	597.9913	0.9353
49	1	761.5804	5	5.3914	50	597.9885	0.9787
50	1	761.608	5	5.3953	100	597.9877	1.006
51	1	761.6194	5	5.3969	250	597.9876	1.0172
52	1	761.6232	5	5.3974	500	597.9875	1.021
53	1	760.8703	8	5.1169	0	601.5279	3.5471
54	1	760.7542	8	5.1355	0.1	600.8223	2.8374
55	1	760.6939	8	5.1525	0.2	600.3082	2.3229
56	1	760.6997	8	5.2162	0.5	599.0368	1.0619
57	1	760.7636	8	5.2402	1	598.734	0.7753
58	1	761.3208	8	5.3527	5	598.037	0.725
59	1	761.3746	8	5.361	8	598.02	0.7766
60	1	761.4183	8	5.3677	10	598.0091	0.8191
61	1	761.5014	8	5.38	20	597.9949	0.9007
62	1	761.5804	8	5.3914	50	597.9885	0.9787

GA BASED MULTIOBJECTIVE ECONOMIC LOAD DISPATCH

63	1	761.6003	8	5.3942	100	597.9879	0.9983
64	1	761.6162	8	5.3965	250	597.9876	1.0141
65	1	761.6216	8	5.3972	500	597.9876	1.0194
66	1	760.9438	10	5.1087	0	601.9011	3.9242
67	1	760.754	10	5.1356	0.1	600.8211	2.8362
68	1	760.6939	10	5.1525	0.2	600.3085	2.3232
69	1	760.6602	10	5.1817	0.5	599.6231	1.6403
70	1	760.7636	10	5.2402	1	598.7339	0.7752
71	1	761.2218	10	5.3366	5	598.0808	0.6337
72	1	761.346	10	5.3566	8	598.0285	0.749
73	1	761.3939	10	5.364	10	598.0148	0.7953
74	1	761.5014	10	5.38	20	597.9949	0.9007
75	1	761.5448	10	5.3863	50	597.9906	0.9435
76	1	761.6003	10	5.3942	100	597.9879	0.9983
77	1	761.6162	10	5.3965	250	597.9876	1.0141
78	1	761.6216	10	5.3972	500	597.9876	1.0194
79	1	761.273	20	5.0856	0	603.2977	5.3455
80	1	761.0845	20	5.0968	0.1	602.5378	4.5702
81	1	760.9498	20	5.1081	0.2	601.93	3.9534
82	1	760.6602	20	5.1817	0.5	599.6227	1.6399
83	1	760.6602	20	5.1818	1	599.6203	1.6375
84	1	761.2215	20	5.3366	5	598.0809	0.6335
85	1	761.2171	20	5.3358	8	598.0833	0.6296
86	1	761.2819	20	5.3464	10	598.0522	0.6883
87	1	761.435	20	5.3702	20	598.0056	0.8355
88	1	761.5448	20	5.3863	50	597.9906	0.9435
89	1	761.5404	20	5.3857	100	597.991	0.9392
90	1	761.5914	20	5.393	250	597.9881	0.9896
91	1	761.6091	20	5.3955	500	597.9877	1.0071
92	1	761.8534	50	5.0667	0	605.3183	7.4273
93	1	761.6582	50	5.0713	0.1	604.6751	6.7617
94	1	761.1675	50	5.0914	0.2	602.8824	4.9212
95	1	761.1675	50	5.0914	0.5	602.8823	4.9211
96	1	760.6602	50	5.1819	1	599.6198	1.637
97	1	760.7787	50	5.2448	5	598.6836	0.7304
98	1	760.9432	50	5.285	8	598.3347	0.5021
99	1	761.0273	50	5.302	10	598.2303	0.5031
100	1	761.2621	50	5.3432	20	598.0609	0.67
101	1	761.3372	50	5.3552	50	598.0314	0.7406
102	1	761.5404	50	5.3857	100	597.991	0.9392
103	1	761.5914	50	5.393	250	597.9881	0.9896

GA BASED MULTIOBJECTIVE ECONOMIC LOAD DISPATCH

104	1	761.6091	50	5.3955	500	597.9877	1.0071
105	1	762.2433	100	5.061	0	606.5345	8.6924
106	1	762.1047	100	5.0626	0.1	606.111	8.2509
107	1	761.9787	100	5.0645	0.2	605.7179	7.842
108	1	761.6644	100	5.0711	0.5	604.6956	6.7829
109	1	761.2974	100	5.0844	1	603.3908	5.4408
110	1	760.6602	100	5.1821	5	599.6151	1.6323
111	1	760.7244	100	5.2267	8	598.8955	0.9257
112	1	760.789	100	5.2478	10	598.6521	0.703
113	1	761.0456	100	5.3055	20	598.2115	0.5098
114	1	761.3372	100	5.3552	50	598.0314	0.7406
115	1	761.2883	100	5.3475	100	598.0495	0.6943
116	1	761.4758	100	5.3763	250	597.9984	0.8755
117	1	761.5485	100	5.3869	500	597.9903	0.9472
118	1	762.5766	250	5.0587	0	607.5214	9.7246
119	1	762.4339	250	5.0595	0.2	607.1038	9.2872
120	1	762.2408	250	5.061	0.5	606.5269	8.6844
121	1	761.2974	250	5.0844	1	603.3907	5.4407
122	1	760.937	250	5.1094	5	601.8679	3.8906
123	1	760.7281	250	5.1416	8	600.6252	2.6399
124	1	760.6755	250	5.1614	10	600.0766	2.0917
125	1	760.7284	250	5.2282	20	598.8762	0.9074
126	1	760.7982	250	5.2504	50	598.6256	0.6806
127	1	761.2883	250	5.3475	100	598.0495	0.6943
128	1	761.4758	250	5.3763	250	597.9984	0.8755
129	1	761.5485	250	5.3869	500	597.9903	0.9472
130	1	762.7114	500	5.0583	0	607.9099	10.1322
131	1	762.6794	500	5.0584	0.1	607.8182	10.0359
132	1	762.632	500	5.0585	0.2	607.6815	9.8925
133	1	762.5189	500	5.059	0.5	607.3533	9.5485
134	1	762.3458	500	5.0601	1	606.8422	9.0137
135	1	761.4411	500	5.0783	5	603.9204	5.9841
136	1	761.0951	500	5.0961	8	602.5827	4.6159
137	1	760.937	500	5.1094	10	601.8679	3.8906
138	1	760.6755	500	5.1614	20	600.0766	2.0917
139	1	760.7982	500	5.2504	50	598.6256	0.6806
140	1	760.7994	500	5.2507	100	598.6222	0.6777
141	1	761.1452	500	5.3236	250	598.1278	0.5704
142	1	761.3506	500	5.3573	500	598.027	0.7534
143	1	762.7842	1000	5.0582	0	608.1175	10.3503
144	1	762.7631	1000	5.0582	0.1	608.0574	10.2871

GA BASED MULTIOBJECTIVE ECONOMIC LOAD DISPATCH

145	1	762.7422	1000	5.0583	0.2	607.9978	10.2245
146	1	762.583	1000	5.0587	1	607.5398	9.7439
147	1	762.3458	1000	5.0601	5	606.8422	9.0137
148	1	761.6332	1000	5.072	8	604.5903	6.6741
149	1	760.9568	1000	5.1074	10	601.9635	3.9874
150	1	760.9556	1000	5.1075	20	601.9577	3.9815
151	1	760.6602	1000	5.1822	50	599.6138	1.631
152	1	760.7994	1000	5.2507	100	598.6222	0.6777
153	1	761.1452	1000	5.3236	250	598.1278	0.5704
154	1	761.3506	1000	5.3573	500	598.027	0.7534
155	1	761.0441	0	5.3052	1	598.213	0.5092
156	1	760.6664	1	5.1684	0	599.91	1.9257
157	0	762.8597	1	5.0582	0	608.3313	10.5751
158	0	762.583	0	5.0587	1	607.5398	9.7439
159	1	760.6602	0	5.1811	0	599.6354	1.6525

6.2 2D GRAPHS:

The results of Table 6.1 are represented by 2D graphs by taking three different combinations of objective functions. The variation of transmission loss (F_L) with the environmental pollution (F_P) is shown in Fig 6.1. Similarly, Fig 6.2 and Fig 6.3 show the variation of cost of generation (F_C) with environmental pollution (F_P) and the variation of cost of generation (F_C) with the transmission loss (F_L) respectively.

(i)Transmission Loss vs Environmental Pollution

The Fig-6.1 represents the variation of transmission loss (F_L) with the variation in environmental pollution (F_P). X-Axis represents the transmission loss of system. Y-Axis represents the environmental pollution.

The curve shows the behaviour F_L & F_P in range (5.0582 to 5.3976 MW) and (597.9875 to 608.3315 kg/hr) respectively. From the curve, when the transmission loss is increased from 5.0582 MW to 5.3976 MW, then the environmental pollution decreases from 608.3315 kg/hr to 597.9875 kg/hr. Therefore, it can be concluded that in the entire range, both the objectives are 'conflicting' in nature. .

(ii)Cost of Generation vs Environmental Pollution

Fig. 6.2 represents the variation of cost of generation (F_C) with the environmental pollution (F_P). X-Axis represents the cost of generation. Y-Axis represents environmental pollution.

The curve A_2P_2 shows that when the environmental pollution takes value from 598 kg/hr to 600 kg/hr, the cost of generation decreases from 761.75 \$/hr to 760.70 \$/hr. Therefore, in this range, the environmental pollution (F_P) & the cost of generation (F_C) are conflicting in nature.

The curve P_2B_2 shows that when the environmental pollution (F_P) takes value from 600 kg/hr to 609 kg/hr, the cost of generation (F_C) also increases from 760.70 \$/hr to 762.88 \$/hr. Therefore in this range both objectives are 'supportive' in nature. Thus it can be concluded that the environmental pollution (F_P) & cost of generation (F_C) may not always follows conflicting behaviour. The objectives can be conflicting in some range & supportive in other range.

(iii) Cost of Generation vs Transmission Loss

The fig. 6.3 represents the variation cost of generation (F_C) with the Transmission Loss (F_L). X – Axis represents the cost of generation (F_C). Y–Axis represents Transmission Loss (F_L) of the system.

The curve A_3P_3 shows that when the Transmission Loss (F_L) increase in range 5.0582 MW to 5.2 MW, the cost of generation (F_C) decreases from (762.583 \$/hr to 760.6602 \$/hr. Thus in this range, the objectives are conflicting in nature.

The curve P_3B_3 show that when the Transmission Loss (F_L) increases in range 5.2 MW to 5.4 MW , the cost of generation (F_C) also increases from 760.6602 \$/hr to 761.62 \$/hr. Thus, in this in range ,the objectives are supportive in nature.

Therefore, it can be concluded from fig. 6.3 that the Cost of generation (F_C) & Transmission Loss (F_L) show both conflicting and supportive behaviour.

6.3 3D GRAPHS:

Fig. 6.4 and 6.5 show the variation of cost of Generation (F_C), Transmission Loss (F_L) and Environmental Pollution (F_P). The curve $A_4P_4B_4$ and $A_5P_5B_5$ show the non-inferior set in 3D space. The region above the curves is feasible region and all solutions within this region are inferior. The region below the curves is infeasible region. The ideal point (760.6602 \$/hr, 5.0582 MW, 597.9875 kg/hr) i.e. (F_{Cmin} , F_{Lmin} , F_{Pmin}) is shown at the origin in 3D graphs and is marked IP. TP is the target point which is at minimum distance from the Ideal Point.

The graph can be divided into two parts- curve A_5P_5 and curve P_5B_5

CURVE A_5P_5 shows that all objectives F_L , F_C , F_P are conflicting in nature.

CURVE P_5B_5 shows that cost of generation (F_C) and environmental pollution (F_P) are supportive in nature. On the other hand transmission loss (F_L) and environmental pollution (F_P) are conflicting in nature. It is therefore concluded that the three objectives may not always follow the traditional behaviour, their behaviour can be different in different ranges.

6.4 DISCUSSION

The Ideal Point i.e. (F_{Cmin} , F_{Lmin} , F_{Pmin}) is located from the Table 6.1 as ($F_{Cmin}=760.6602$ \$/hr, $F_{Lmin}= 5.0582$ MW, $F_{Pmin} =597.9875$ kg/hr). It is observed that in case of 3D individual minimization of an objective function may not give its minimum value. Therefore, Ideal Point has to be located from the observations of Table 6.1. The last column (distance) shows the distance of each of the noninferior solution from the Ideal point. The distance is calculated by using eq (4.2). The Target point is one for which this distance is minimum. It is highlighted at S.No 44 in Table 6.1.

2D and 3D graphs have been plotted from the computational result of table 6.1. 2D graphs are shown by fig 6.1, 6.2, and 6.3 for two objectives. It can be clearly observed from these graphs that the objective follow different behaviour in different domains. It means that they may not always be ‘conflicting’ or ‘supportive’ in all the domains or ranges of objective functions. The objective can be ‘conflicting’ in one range and can follow the supportive behaviour in the other range. Fig 6.4 and 6.5 show the noninferior set in 3D space. In these graphs, the origin represents the Ideal Point i.e (F_{Cmin} , F_{Lmin} , F_{Pmin}). The Target Point is market as TP and is seen closest to the origin (Ideal Point) for IEEE 5 bus system. 3D curve is also drawn in fig 6.4.

However it can be realized that the Ideal Point as the defined and located above cannot be achieved in practice its means that there cannot be any choice of weights which can give the minimum value of all objective function at the same time therefore, one can at the most achieve a point which is a feasible point and at minimum distance from the ideal point has been calculated and depicted in column 7 of table 6.1 for IEEE 5 bus system. The point which is at the minimum distance from the ideal point is highlighted in the S.No. 44, giving the following values for the individual objective function for IEEE 5 bus system

$$F_C = 760.9929 \text{ \$/hr.}$$

$$F_L = 5.2952 \text{ MW.}$$

$$F_P = 598.2692 \text{ kg/hr.}$$

CHAPTER 7

CONCLUSIONS AND FUTURE SCOPE

7.1 CONCLUSIONS

In this work, Multiobjective Economic Load dispatch (MOELD) problem has been formulated using weighting method in 3D space using Genetic Algorithm.

The focus of this thesis is on simultaneous minimization of three objectives of power system— Cost of generation, Transmission loss and Environmental pollution. Multiobjective Economic Load dispatch (MOELD) problem has been formulated by using weighting method. The noninferior set for IEEE 5 Bus system has been obtained by parametrically varying weights attached to the objectives. MOELD problem has been solved by GA tool of MATLAB and the Target Point (TP) or best compromise solution is obtained by using Ideal Distance Minimization Method. 2D and 3D graphs have been plotted from the computational results. It is clearly observed from the graphs that the objectives follow different behaviour in different domains. It means that they may not always be ‘conflicting’ or ‘supportive’ in all the domains or ranges of objective functions. The objective can be ‘conflicting’ in one range and can follow the supportive behaviour in the other range.

7.2 SCOPE FOR FUTURE WORK

In addition to cost of generation, system transmission loss and environmental pollution, other objectives like security, reliability can also be considered.

Multiobjective Economic load dispatch (MOELD) problem can be solved using different methods. There are various techniques for generating noninferior solutions, e.g. Constraint method, noninferior estimation (NISE) method and step method (STEM). These methods generate noninferior set in 3D space for IEEE 5, 14 and 30 bus systems. Approximation of the noninferior set is the most desirable feature for practical problems.

APPENDIX

IEEE 5 BUS SYSTEM

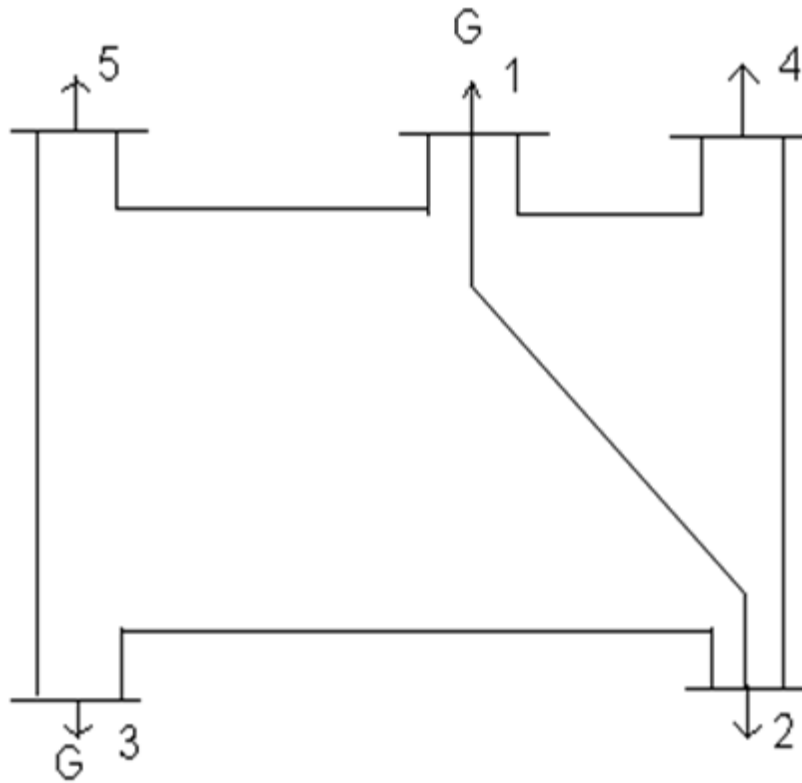


Fig. Bus-Code Diagram of IEEE 5 Bus System

TABLE I

LINE DATA or IMPEDANCE DATA (5 Bus System)

LINE DESIGNATION	*R (p.u.)	*X (p.u.)	LINE CHARGING
1-2	0.10	0.4	0.0
1-4	0.15	0.6	0.0
1-5	0.05	0.2	0.0
2-3	0.05	0.2	0.0
2-4	0.10	0.4	0.0
3-5	0.05	0.2	0.0

* The impedances are based on MVA as 100.

TABLE II

BUS DATA or OPERATING CONDITIONS (5 Bus System)

BUS NO.	GENERATION	GENERATION	LOAD	LOAD
	MW	VOLTAGE MAGNITUDE	MW	MVAR
1*	-----	1.02	-----	-----
2	-----	-----	60	30
3	100	1.04	-----	-----
4	-----	-----	40	10
5	-----	-----	60	20

*Slack Bus

TABLE III

REGULATED BUS DATA (5 Bus System)

BUS NO.	VOLTAGE MAGNITUDE	MINIMUM MVAR CAPABILITY	MAXIMUM MVAR CAPABILITY	MINIMUM MW CAPABILITY	MAXIMUM MW CAPABILITY
1	1.02	0.0	60	30	120
3	1.04	0.0	60	30	120

The nodal load voltage inequality constraints are $0.9 \leq V_i \leq 1.05$

Cost Characteristics

The cost characteristics of the IEEE 5 Bus System are as follows:

$$C_1 = 50 P_1^2 + 351 P_1 + 44.4 \text{ \$/hr}$$

$$C_3 = 50 P_3^2 + 389 P_3 + 40.6 \text{ \$/hr}$$

Here, the total load demand of the system is 165 MW. Maximum and minimum active power constraint on the generator bus for the given system is 120 MW and 50 MW respectively.

Voltage magnitude constraint for generator at bus 3 is 1.04 pu.

Emission Characteristics

$$E_1 = 126 P_1^2 - 135.5 P_1 + 22.983 \text{ \$/hr}$$

$$E_3 = 76.5 P_3^2 + 80.5 P_3 + 363.704 \text{ \$/hr}$$

M-file For Calculating B- Coefficients:

clear

basemva=100;

accuracy=0.0001;

maxiter=10;

```
busdata=[1 1 1.02 0 0 0 0 0 0 60 0;2 0 1 0 60 30 0 0 0 0 0;3 2 1.04 0 0 0 82 0 0 60  
0;4 0 1 0 40 10 0 0 0 0 0;5 0 1 0 60 20 0 0 0 0 0];
```

```
linedata=[1 2 0.10 0.4 0 1;1 4 0.15 0.6 0 1;1 5 0.05 0.2 0 1;2 3 0.05 0.2 0 1;2 4 0.10  
0.4 0 1;3 5 0.05 0.2 0 1];
```

```
disp(busdata)
```

```
disp(linedata)
```

```
mwlimits=[30 120;30 120];
```

```
lfybus
```

```
lfnewton
```

```
busout
```

```
bloss
```

B- Coefficients Calculated (based on equal intercept criterion)

$$B_{11} = 0.00035336$$

$$B_{12} = 0.0000206392$$

$$B_{21} = 0.0000206392$$

$$B_{22} = 0.0003689$$

M- File FOR 3-D PROBLEM:

Objective Function M-file:

Function $z = \text{objective5bus}(x)$

$$z = ((50*(x(1) / 100)*(x(1) / 100)) + (351*(x(1) / 100)) + 44.4 + (50*(x(3) / 100)*(x(3) / 100)) + (389*(x(3)/100)) + 40.6);$$

Constraint Function M-file:

Function $[c, \text{ceq}] = \text{constraint14bus}(x)$

$$c = [-x(1)+50; x(1)-120; -x(3)+50; x(3)-120];$$

$$\text{ceq} = (x(1)+x(3))*((0.00035336*x(1)*x(1)) + 0.000206392*x(1)*x(3)) + (0.0003689*x(3)*x(3)) - 165);$$

REFERENCES

- [1] Wadhwa C L, Jain N K. 'Multiple objective optimal load flow: a new perspective.' IEE Proceeding, C, 1990, 137(1), pp. 11-18.
- [2] Nangia U, Jain N K, Wadhwa C L. 'Noninferior Set Estimation for Multiobjective Optimal Load Flow Study.' IE(I) Journal-EL, Vol. 78, 1998.
- [3] Cohon J L. 'Multiobjective programming and planning.' Academic Press, London 1978.
- [4] Abido, M A. 'A novel Multiobjective Evolutionary Algorithm for environmental economic power dispatch. Electric power systems research.' Vol. 65, Issue 1, April 2003, pp. 71-81.
- [5] Nangia U, Jain N K, Wadhwa C L. 'Multiobjective optimal load flow based on ideal distance minimization in 3D space.' Electrical power and energy systems (23), 2001, pp. 847-855.
- [6] Negnevitsky, M. 'Artificial Intelligence: a guide to intelligence system.' Essex, England: Addison-Wesley 2002.
- [7] Santos J R et al. 'Finding improved local minima of power system optimization problems by interior-point methods.' IEEE transactions on power systems, Vol. 18, NO. 1, Feb. 2003, pp. 238-244.
- [8] Sheble G B, Brittig K. 'Refined genetic algorithm economic dispatch example.' IEEE transaction on power systems, Feb 1995. Vol. 10, NO. 1, pp. 117-124.
- [9] Walters D C, Sheble G B. 'Genetic algorithm solution of economic dispatch with valve point loading.' IEEE transactions on power systems, Aug 1993, Vol. 8 Issue 3, pp. 1325-1332.
- [10] Kumaran G and Mouly VSRK. 'Using evolutionary computation to solve the economic load dispatch problem.' Proceedings of the 2001 congress on evolutionary computation, Vol. 1, pp. 296 – 301.
- [11] Chiang C L. 'Genetic algorithms for power economic.' IET on generation, transmission and distribution, 2007, Vol. 1, No.2, pp. 261-269.

[12] Zhang et al. 'A new hybrid real-coded genetic algorithm and application in dynamic economic dispatch.' IEEE proceeding on intelligence control and automation, June 2006, Dalian, China, pp. 3627-3632.

[13] Wong K P and Wong Y W. 'Genetic and genetic simulated annealing algorithms approach to economic dispatch.' IEEE proceeding on Generation, Transmission and Distribution, 1994, Vol. 141, No.5, pp. 507-512.

[14] Das D B and Patvardhan C. 'New multiobjective stochastic search technique for economic load dispatch.' IEEE proceeding on Generation, Transmission and Distribution, 1998, Vol. 145, No.6, pp. 747-752.

[15] Chiang C L et al. 'Improved genetic algorithm for economic dispatch of power system having special units.' IEEE Conference on Cybernetic and Intelligence Systems, 2006, pp. 1-6.

[16] Chen P H and Chang H C. 'Large-scale economic dispatch by genetic algorithm.' IEEE transactions on power systems, Nov 1995, Vol. 10, Issue. 4, pp. 1919-1926.

[17] Nanda J and Narayanan R B. 'Application of genetic algorithm to economic load dispatch with line flow constrain.' International Electrical Power & Energy Systems, 2002, Vol. 24, pp. 723-729.

[18] Ongsakul W and Ruangpayoongsak R. 'Constrained economic dispatch by simulated annealing genetic algorithms.' IEEE power engineering society international conference on power industry computer application, 2001, pp. 207-212.

[19] Sudhakaran M and Slochanal DSMR. 'Integrating genetic algorithms and tabu search for combined heat and power economic dispatch.' Proceeding of Conference on Convergent Technologies for Asia-Pacific Region, 2003, Vol. 1, pp. 67-71.

[20] Kumarappan N and Mohan R. 'Neuro-hybrid genetic algorithms based economic dispatch for utility system.' Proceedings of the International Joint Conference on Neural Networks, Vol. 3, pp. 2112-2117.

[21] Wang P Y et al. 'Fuzzy logic controlled genetic algorithm.' Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, Sep 1996, Vol. 2, pp. 972-979.

- [22] Edgeworth F Y. 'Mathematical Physics.' P Keagan, London, England, 1881.
- [23] V Pareto. 'Cours D'Economie Politique, vol. I and II.' F Rouge, Lausanne, 1896.
- [24] K M Miettinen, 'Nonlinear Multiobjective Optimization.' Kluwer Academic Publishers, Boston, Massachusetts, 1999.
- [25] M Ehrgott, 'Multicriteria Optimization.' Springer, second edition, 2005.
- [26] C A Coello Coello and A Hernández Aguirre. 'Design of Combinational Logic Circuits through an Evolutionary Multiobjective Optimization Approach.' Artificial Intelligence for Engineering, Design, Analysis and Manufacture, Vol. 16, no. 1, Jan. 2002, pp. 39–53.
- [27] M T Jensen. 'Guiding single-objective optimization using multi-objective methods.' In Günther Raidl et al, editor, Applications of Evolutionary Computing. Evoworkshops. 2003: EvoBIO, EvoCOP, EvoIASP, EvoMUSART, EvoROB, and EvoSTIM, Essex, UK, Springer. Lecture Notes in Computer Science Vol. 2611, Apr. 2003, pp.199–210.
- [28] J D Knowles, R A Watson, and D W Corne. 'Reducing Local Optima in Single-Objective Problems by Multi-objectivization.' In Eckart Zitzler et al, editor, First International Conference on Evolutionary Multi-Criterion Optimization, Springer-Verlag. Lecture Notes in Computer Science Vol. 1993, 2001, pp. 268–282.
- [29] J Kennedy and R C Eberhart. 'Swarm Intelligence.' Morgan Kaufmann Publishers, San Francisco, California, 2001.
- [30] K V Price. 'An Introduction to Differential Evolution.' In David Corne, Marco Dorigo, and Fred Glover, editors, New Ideas in Optimization, McGraw-Hill, London, UK, 1999, pp. 79–108.
- [31] H A Abbass and R Sarker. 'The pareto differential evolution algorithm.' International Journal on Artificial Intelligence Tools, Vol. 11, No. 4, 2002, pp. 531–552
- [32] C A Coello Coello, G Toscano Pulido, and M Salazar Lechuga. 'Handling multiple objectives with particle swarm optimization.' IEEE Transaction, Evolutionary Computation, Vol. 8, No. 3, June 2004, pp. 256–279.

- [33] C A Coello Coello and N Cruz Cortés. ‘Solving multiobjective optimization problems using an artificial immune system.’ *Genetic Programming and Evolvable Machines*, Vol. 2, June 2005, pp. 163–190.
- [34] M Guntsch. ‘Ant Algorithms in Stochastic and Multi-Criteria Environments.’ Ph.D. thesis, Department of Economics and Business Engineering, University of Karlsruhe, Germany, 2004.
- [35] D Buche, S Muller and P Koumoutsakos. ‘Self-adaptation for multi-objective evolutionary algorithms.’ In Carlos M Fonseca et al, editor, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003, Faro, Portugal, Springer. Lecture Notes in Computer Science*, Vol. 2632, Apr. 2003, pp. 267–281.
- [36] J Knowles and E J Hughes. ‘Multiobjective optimization on a budget of 250 evaluations.’ In Carlos A Coello Coello et al, editor, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005, Guanajuato, México, Springer. Lecture Notes in Computer Science*, Vol. 3410, Mar. 2005, pp. 176–190.
- [37] Kothari D P and Dhillon J S. ‘Power System Optimization, Prentice Hall of India.’ 2006.
- [38] Chowdhury B H. and Rahman S. ‘A review of recent advances in economic dispatch.’ *IEEE transaction on power system*, Nov. 1990, Vol. 5, No. 4, pp. 1248-1259.
- [39] Deschamps D. ‘Optimization in power system planning.’ In El-Abiad, AH. Ed. *Power system analysis and planning*, London, Hemishpere Publishing Corporation, 1981, pp. 201-203.
- [40] Deschamps D. ‘Optimization in power system planning.’ In El-Abiad, AH. Ed. *Power system analysis and planning*, London, Hemishpere Publishing Corporation, 1981, pp. 203-205.
- [41] Deschamps D. ‘Optimization in power system planning.’ In El-Abiad, AH. Ed. *Power system analysis and planning*, London, Hemishpere Publishing Corporation, 1981, pp. 205-208.
- [42] Lee K Y et al. ‘Adaptive hopfield neural network for economic dispatch.’ *IEEE transactions on power systems*, Vol. 18, NO. 2, Feb. 2003 pp. 519-529.

- [43] Hong Y Y and Li C L. 'Genetic algorithms based economic dispatch for cogeneration units considering multiplant multibuyer wheeling.' IEEE Transactions on Power Systems, Feb 2002, Vol. 17, Issue. 1, pp. 134-140.
- [44] Hosseini S H and Kheradmandi M. 'Dynamic economic dispatch in restructured power systems considering transmission costs using genetic algorithm.' IEEE on electrical and computer engineering, Canadian Conference, Vol. 3, 2-5 May 2004, pp. 1625 – 1628.
- [45] Hong Y Y and Li C T. 'Short term real power scheduling considering fuzzy factor in an autonomous system using GA.' IEEE Transaction on generation, transmission and distribution, 2006 ,Vol. 152, No.6, pp. 684-691.
- [46] Ma H et al. 'Genetic algorithms based approach to economic dispatch of power systems.' IEEE Proceeding on Shouteastcon '94 'creative technology transfer – a global affair', 1994, pp. 212-216.
- [47] Yalcinoz T and Altun H. 'Environmentally economic dispatch via genetic algorithm with arithmetic crossover.' IEEE Afrcon 2002, pp. 923-927.
- [48] Goldberg D E. 'Genetic Algorithms in Search, Optimization and Machine Learning.' Addison-Wesley, Reading, MA, 1989.
- [49] Gen M and Cheng R. 'Genetic Algorithms and Engineering Design.' Wiley, New York, 1997.
- [50] Albrecht R, Reeves C, and Steele N. 'Artificial Neural Nets and Genetic Algorithms.' Springer-Verlag, New York, 1993.
- [51] Stevenson W D, Grainger J J. 'Power System Analysis.' Tata McGraw-Hill Book Company, New York, USA, 2003, pp. 543-548.
- [52] Singh L and Dhillon J S. 'Surrogate worth trade-off method for economic– emission dispatch.' Power system conference: Advance Metering, protection, control, communication and distributed resources, 2006, pp. 230-235.

[53] EL-Keib A A, Ma H and Hart J L. 'Economic dispatch in view of clean air act of 1990.' IEEE transaction on power systems, Vol.9, No.3, May 1994, pp. 972-978.