

## CHAPTER 1

### INTRODUCTION

#### **1.1 OVERVIEW**

The optimal power system operation is achieved when the best values of various objectives of power system viz. cost of generation, system transmission losses and environmental pollution etc. are simultaneously attained. But 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 intolerably bad. But there is no other solution to facilitate the decision making process. 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 14 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 14 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 six 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: Discusses the study of GATool Box by varying different parameters.

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 14 bus system.

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

<b>GA Terms</b>	<b>Corresponding Optimization Description</b>
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].

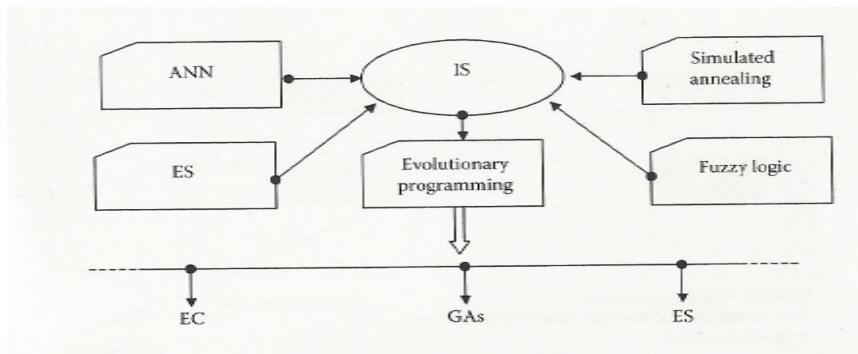


Fig 2.1 Common Classifications of IS

### 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.

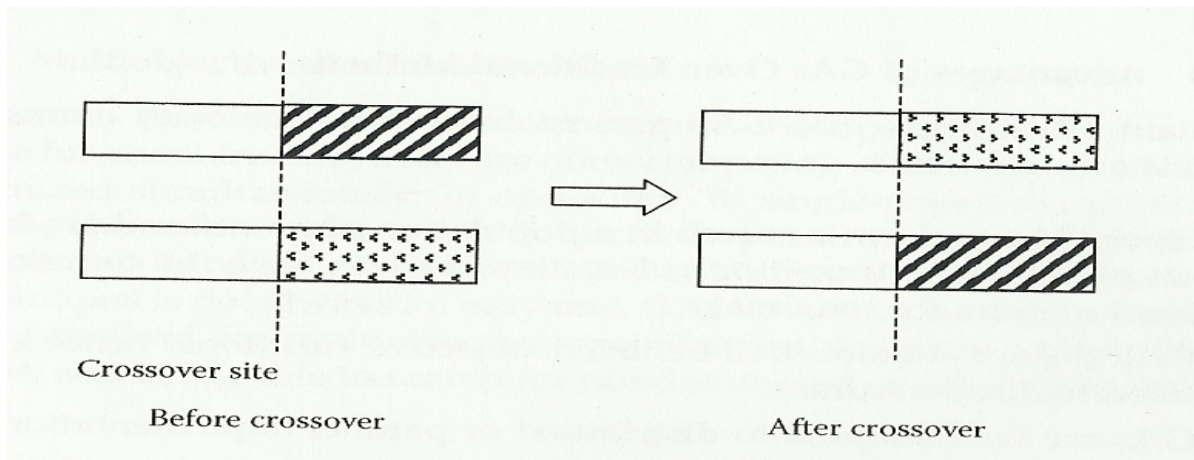


Fig 2.2 Crossover operation on a pair of strings

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**

### **OPTIMIZATION USING GA TOOLBOX**

#### **3.1 EXPLORING THE TOOLS OF GA**

In Matlab, GA operations are explored in many ways to get the global optimal points: running the GA from the command prompt or executing the GA tool in a separate window.

##### **3.1.1 Calling the Function at the Command Line**

To use the genetic algorithm at the command line, call the genetic algorithm function `ga` with the syntax:

```
[x fval] = ga(@fitnessfun, nvars, options)
```

Where,

@fitnessfun - A function handle to the M-file that computes the fitness function.

nvars -the number of independent variables for the fitness function.

If it does not pass in this argument, GA uses its default options.

The output results are given by

x — Point at which the final value is attained

fval — Final value of the fitness function at x

Return results directly to the MATLAB® workspace.

Run the genetic algorithm multiple times with different options, by calling GA from an M-file by clicking file open command on Matlab R2008b.

### 3.1.2 Running the Problem in GA TOOL

To launch the Optimization Tool, go to MATLAB **Start** menu as pictured:

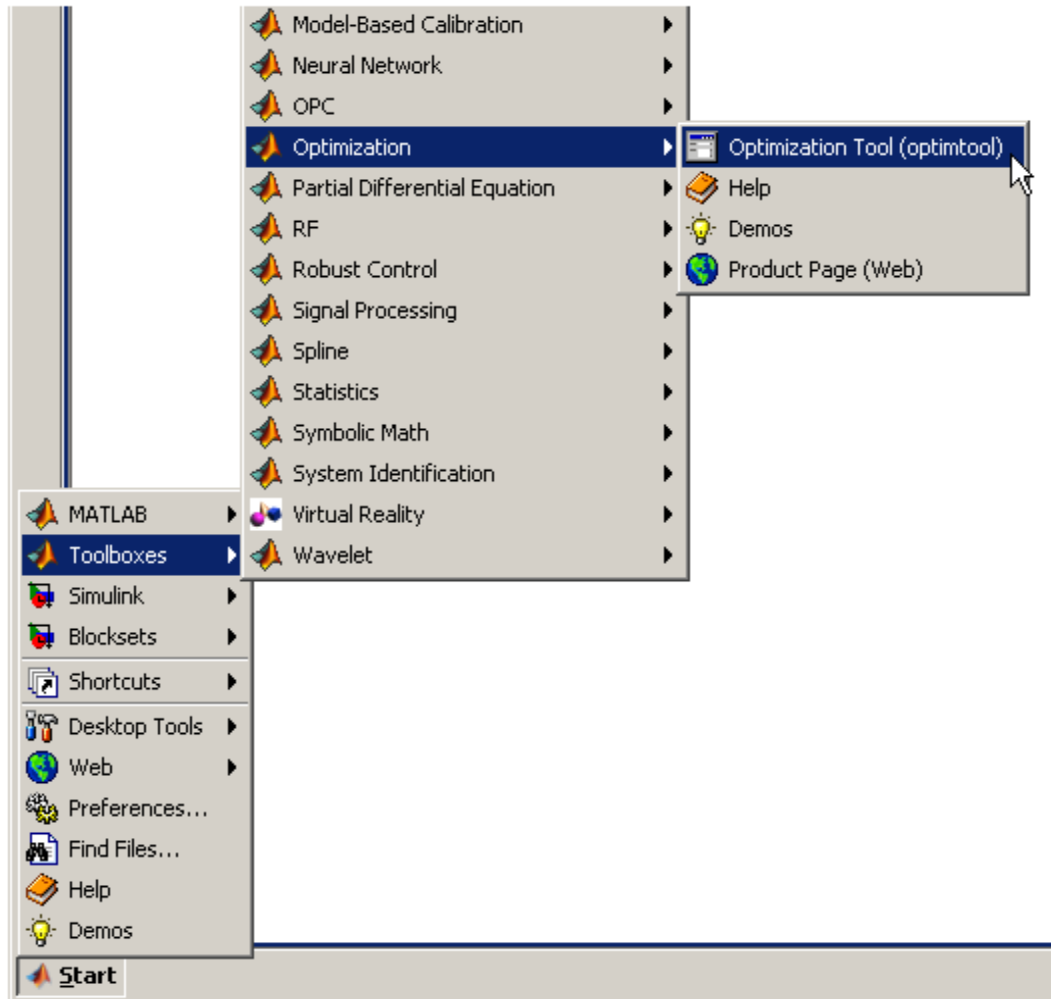
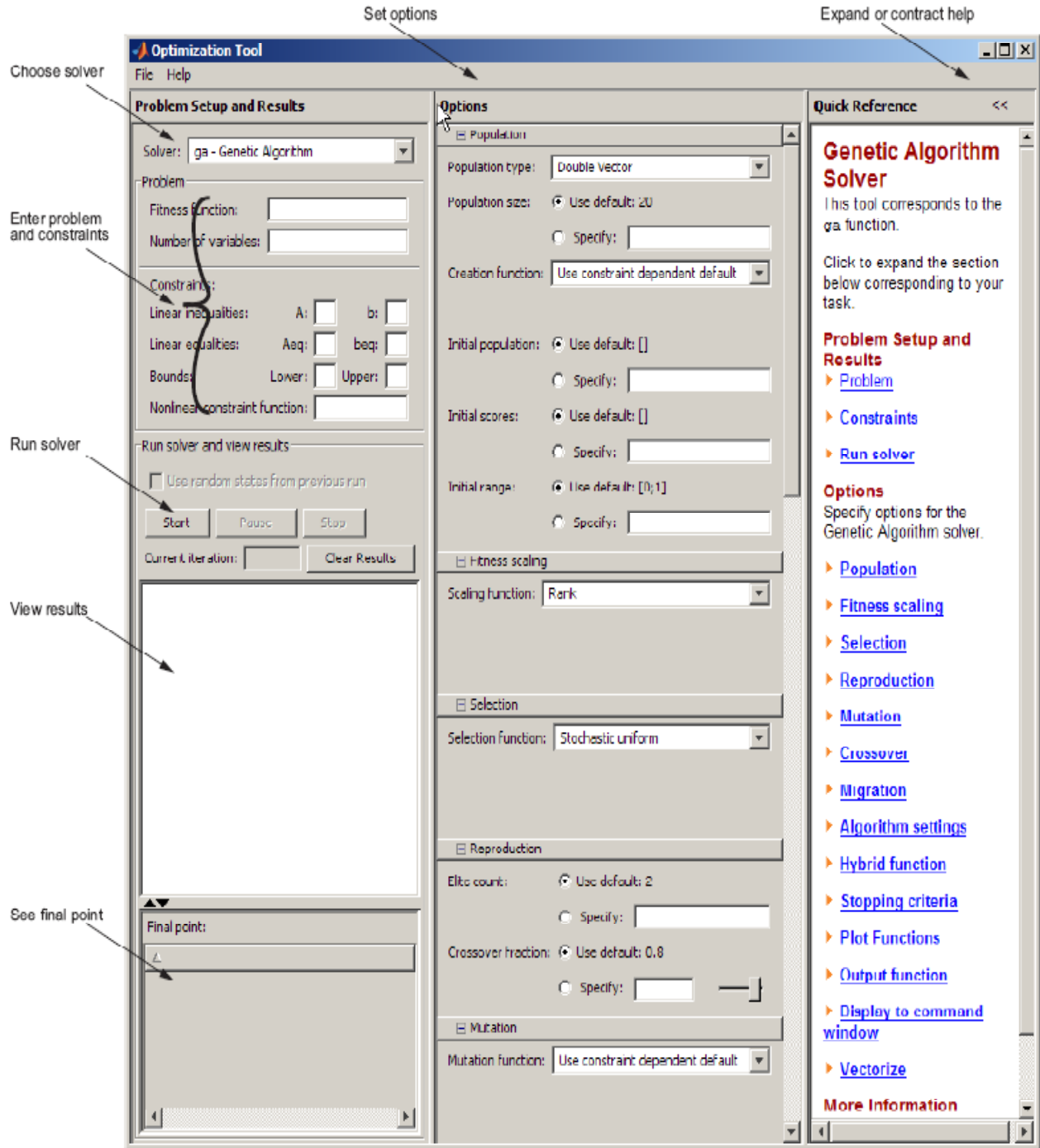


Fig 3.1 Launching Optimization Tool

Another mode to open the optimization tool is to enter `optimtool ('ga')` at the command line, or enter `optimtool` and then choose `ga` from the Solver menu.



You can also launch the tool from the MATLAB Start menu as pictured:

**Fig: 3.2 GA tool box in Matlab**

## 3.2 DIFFERENT PARAMETERS IN GA TOOL

The different parameters of GA toolbox are:-

- Population
- Fitness scaling
- Selection
- Reproduction
- Mutation
- Crossover
- Stopping criteria
- Display to the command window

### 3.2.1 Population

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

Population type specifies the type of the input to the fitness function. It is easy to set Population type to be double vector, or Bit string, or Custom. If we select Custom, then it is necessary to write our own creation, mutation, and crossover functions and specify these functions in the fields Creation function, Mutation function, and Crossover function, respectively.

\*Matlab R2008b uses the default population type as double vector, whereas all the standard textbooks use Bit string as the population type.

Initial population enables us to specify an initial population for the genetic algorithm. If initial population is not specified then, the algorithm creates one using the Creation function. Initial score enables us to specify scores for initial population. If we do not specify initial scores, the algorithm computes the scores using the fitness function. Moreover, Initial range specifies lower and upper bounds for the entries of the vectors in the initial population. It is easy to specify Initial range as a matrix with 2 rows and Initial length columns.

### 3.2.2 Fitness Scaling

The scaling function converts raw fitness scores returned by the fitness function to values in a range that is suitable for the selection function.



Rank scales the raw scores based on the rank of each individual, rather than its score. The rank of an individual is its position in the sorted scores. The rank of the fittest individual is 1, the next fittest is 2, and so on. Rank fitness scaling removes the effect of the spread of the raw scores.

### 3.2.3 Selection

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

It is easy to specify the function that performs the selection in the Selection function field.

The most popular Roulette wheel selection (*mostly used*) is explained below

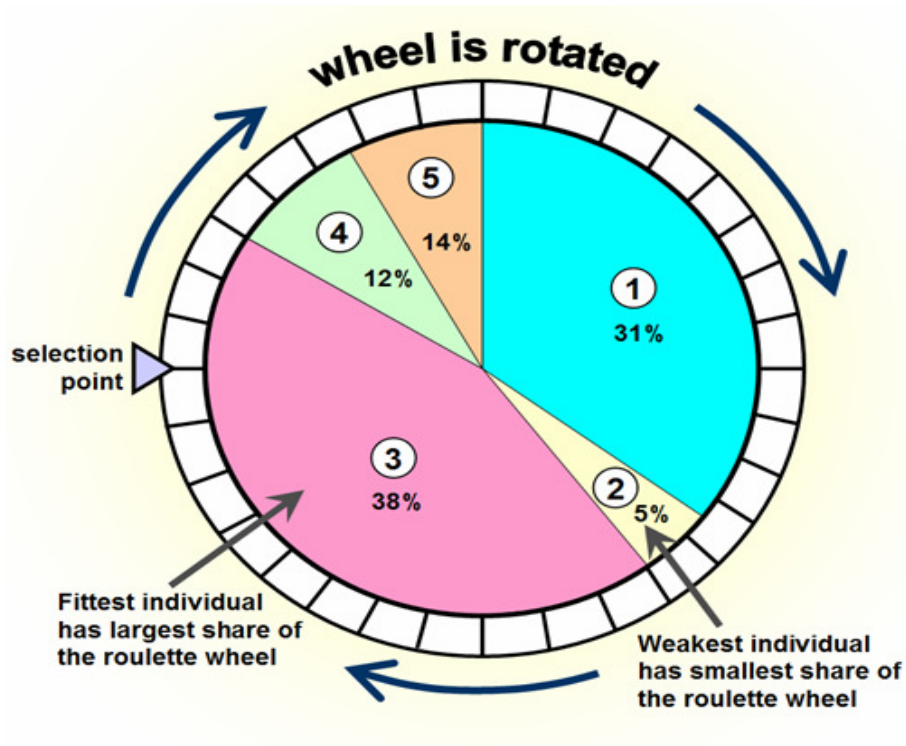


Fig 3.3 Roulette wheel

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.

### 3.2.4 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. Set Elite count to be a positive integer less than or equal to Population size. Crossover fraction specifies the fraction of the next generation, other than elite individuals, that are produced by crossover.

### 3.2.5 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. It is easy to specify the function that performs the mutation in the Mutation function field. The default option in Mutation function field is **Gaussian**. Gaussian is normally used for unconstrained problems. For constrained problems **adapt feasible** option is used.

### 3.2.6 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.

### 3.2.7 Display option

Level of display specifies the amount of information displayed in the MATLAB Command Window when you run the algorithm. Choose from the following:

**Off** — Display no output.

**Iterative** — Display information at each iteration of the algorithm.

**Diagnose** — Information is displayed at each iteration. In addition, the diagnostic lists some problem information and the options that are changed from the defaults.

**Final** — Display only the reason for stopping at the end of the run.

## 3.3 RUN SOLVER IN GA TOOL

To run the solver, click Start under Run solver. When the algorithm terminates, the Status and results pane displays the reason the algorithm terminated. The Final point updates to show the coordinates of the final point.

## 3.4 OPTIMIZATION PROBLEM

### Minimize a fitness function using GA toolbox

This is a demonstration of how to create and minimize fitness function of unconstrained problem with the help of genetic algorithm (GA).

### Unconstrained Minimization Problem

Here we want to minimize an objective function of two variables

Rosenbrock Function:

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

Range  $x_1 \in (0, 2)$ ;  $x_2 \in (0, 2)$

where,

$x_1$  =first variable in objective function

$x_2$  =second variable in objective function

$f(x)$  =value of function

### **Coding the Fitness Function**

We create an M-file named fitness\_var.m with the following code in it:

```
function y = fitness_var(x)
```

The Genetic Algorithm solver assumes the objective function will take one input x, where x is a row vector with as many elements as number of variables in the problem. The fitness function computes the value of the function and returns that scalar value in its one return argument y.

### **M-file of Rosenbrock 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)
```

Rosenbrock function is minimized in the Optimtool by varying various parameters of Genetic Algorithm:-stall generation, function tolerance and population size etc. Tables 3.1, 3.2 and 3.3 shows the results when one of these parameters is varied while keeping the other two at a fixed value.

Table 3.1 shows the results of minimizing the Rosenbrock function when stall generation is varied from 50 to 350 in steps of 25 while population size and function tolerance are fixed to 20 and  $10^{-6}$  respectively.

Table 3.2 shows the results of minimizing the function when function tolerance is varied from  $10^{-1}$  to  $10^{-20}$  in steps of  $10^{-1}$  while stall generation is fixed at 50 and population size is fixed at 20.

Similarly, Table 3.3 shows the results of varying the population size from 4 to 28 in steps of 2 keeping function tolerance fixed at  $10^{-6}$  and stall generation at 50.

**Table 3.1**

RESULTS OF VARIATION OF STALL GENERATION KEEPING :

POPULATION SIZE=20 AND FUNCTION TOLERANCE=1.00E-06

S.no	Stall iteration	$x_1$	$x_2$	F(x)	No. of iterations
1	50	1.080	1.166	0.0067645	51
2	75	0.996	0.980	0.0145548	46
3	100	0.964	0.928	0.0013466	100
<b>4</b>	<b>125</b>	<b>0.938</b>	<b>0.879</b>	<b>0.0041093</b>	<b>76</b>
5	150	1.027	1.053	0.0009641	150
6	175	0.973	0.947	0.0007132	54
7	200	1.037	1.075	0.0013465	76
8	225	0.982	0.966	0.0004320	225
9	250	0.987	0.974	0.0002150	250
10	275	1.004	1.008	0.0002071	275
11	300	0.988	0.976	0.0002262	300
12	325	0.998	0.995	0.0001062	325
13	350	1.024	1.046	0.0007721	350

**Table 3.2**

RESULTS OF VARIATION OF FUNCTION TOLERANCE KEEPING :

POPULATION SIZE=20 AND STALL GENERATION =50

S.no	Function tolerance value	$x_1$	$x_2$	F(x)	No of iterations
1	1.00E-01	1.135507369	1.28949244	0.01836358	51
2	1.00E-02	1.149956617	1.316282653	0.026229451	51
3	1.00E-03	0.920565415	0.839899941	0.011996133	51
4	1.00E-04	0.808898789	0.654957078	0.03656061	51
<b>5</b>	<b>1.00E-05</b>	<b>0.969239844</b>	<b>0.918222241</b>	<b>0.033124841</b>	<b>51</b>
6	1.00E-06	0.600089858	0.357474486	0.160621576	51
7	1.00E-07	0.602569686	0.357099716	0.161539476	51
8	1.00E-08	0.927063295	0.857973913	0.005540012	67
9	1.00E-09	0.794459653	0.644955846	0.061262432	55
10	1.00E-010	1.032945646	1.165314069	0.001360001	51
11	1.00E-011	0.908547828	0.825217524	0.008369338	100
12	1.00E-012	0.882445734	0.764815929	0.045900023	51
13	1.00E-013	0.911667323	0.829699624	0.008009355	55
14	1.00E-014	0.999191861	0.994212083	0.001741454	82
15	1.00E-015	0.879740922	0.77195636	0.014857353	100
16	1.00E-016	0.840322996	0.695191222	0.037490313	51
17	1.00E-017	0.816833205	0.659376507	0.039696599	73
18	1.00E-018	0.80863667	0.660700221	0.041253391	76
19	1.00E-019	1.260071657	1.593927708	0.071415985	100
20	1.00E-020	1.079382027	1.169816826	0.008558958	85

**Table 3.3**

RESULTS OF VARIATION OF POPULATION SIZE KEEPING :

FUNCTION TOLERANCE =1.00E-06 AND STALL GENERATION =50

S.no	Population size	$x_1$	$x_2$	F(x)	No of iterations
1	4	0.350548796	0.083979944	0.573142994	51
2	6	0.828557931	0.731136354	0.228559197	51
3	8	0.822990587	0.667460159	0.041041177	67
4	1	1.086424503	1.18900704	1.086424503	100
5	12	1.014413084	1.030390816	0.000392002	100
6	14	0.705335926	0.500189839	0.087551102	100
7	16	0.977969717	0.959365004	0.001349832	100
8	18	1.000384592	1.002770266	0.000401001	100
<b>9</b>	<b>20</b>	<b>0.919020536</b>	<b>0.849162626</b>	<b>0.008640573</b>	<b>53</b>
10	22	1.004966872	1.010403122	0.000044400	100
11	24	0.967278	0.936299507	0.001115992	100
12	26	1.030931976	1.061720015	0.001077946	100
13	28	0.801663259	0.630511511	0.054105715	100

### 3.5 DISCUSSION

**3.5.1** - In Table 3.1, stall generation is varied in range 50-350 while keeping function tolerance value and population size at default value i.e.  $10^{-6}$  and 20 respectively.

From the Table 3.1 the values of  $x_1$  and  $x_2$  are close to optimum point (1,1), during various values of stall generation but the last column shows that the entries corresponding to more than 200 stall generation value leads to immature termination, that is, the points obtained after the termination process may or may not minimize the fitness function, that's why these values are discarded .

In the stall generation range 50 – 100, values corresponding to 75, 125 and 175 are selected due to mature termination. Further, comparison between 75, 125 and 175, values of  $x_1$  and  $x_2$  are more accurate with stall generation value 125 which is highlighted in Table 3.1

**3.5.2**- Function tolerance value is varied between  $10^{-1}$  to  $10^{-20}$  and keeping stall generation (equal to 50) & population size (equal to 20) constant.

The number of iterations are constant and equal to 51 for a function tolerance range  $10^{-1}$  to  $10^{-7}$ , but the values of  $x_1$  and  $x_2$  are having significant deviation from the optimum point (1,1).

In function Tolerance range  $10^{-8}$  to  $10^{-20}$  excluding  $10^{-10}$  and  $10^{-12}$ , the number of iterations are increased, But the values of  $x_1$  and  $x_2$  are having small deviation from optimum point (1,1) as compared to when the tolerance range is varied from  $10^{-1}$  to  $10^{-7}$ . Also graphically and in practical world, tolerance value up to  $10^{-8}$  is admissible. That's why range of  $10^{-1}$  to  $10^{-8}$  is selected. Now comparing the values of  $x_1$  and  $x_2$  in this range, we find the values corresponding to  $10^{-5}$  are more close to (1, 1). That's why function tolerance value  $10^{-5}$  is used for specified stall generation and population size which is also highlighted in table 3.2. Parameter function tolerance value  $10^{-14}$  is not selected, although corresponding  $x_1$  and  $x_2$  values are accurate, because it is taking more iterations compared to number of iterations when function tolerance value was  $10^{-5}$  which leads to more time consumption.

**3.5.3**- In Table 3.3 population size is varied in range 4 to 28 while keeping stall generation and function tolerance value at default values that is 50 and  $10^{-6}$  respectively. For the population size 4, 6, 8 and 28, the values of  $x_1$  and  $x_2$  are quite far away from the optimum point (1,1). In the population size range 10 to 26, the values of  $x_1$  and  $x_2$  are very close to the optimum point (1,1)



except for the population size of 14. It is observed that for the population size 20, there is small deviation in the value of  $x_1$  and  $x_2$  from the optimum point (1,1) but the number of iterations are 53. Whereas, all other population sizes from 10 to 28 are taking 100 iterations. The S.no 9 corresponding to population size 20 has been highlighted as it takes minimum iterations to achieve the optimum point with reasonable accuracy.

### **3.6 CONCLUSION:**

The best values of different parameters for the problem under consideration are:

Stall generation: 125

Population size: 20

Function tolerance: 1.00E-05

## 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 14 bus system in 3D space. The data of IEEE 14 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 form 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 through 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.

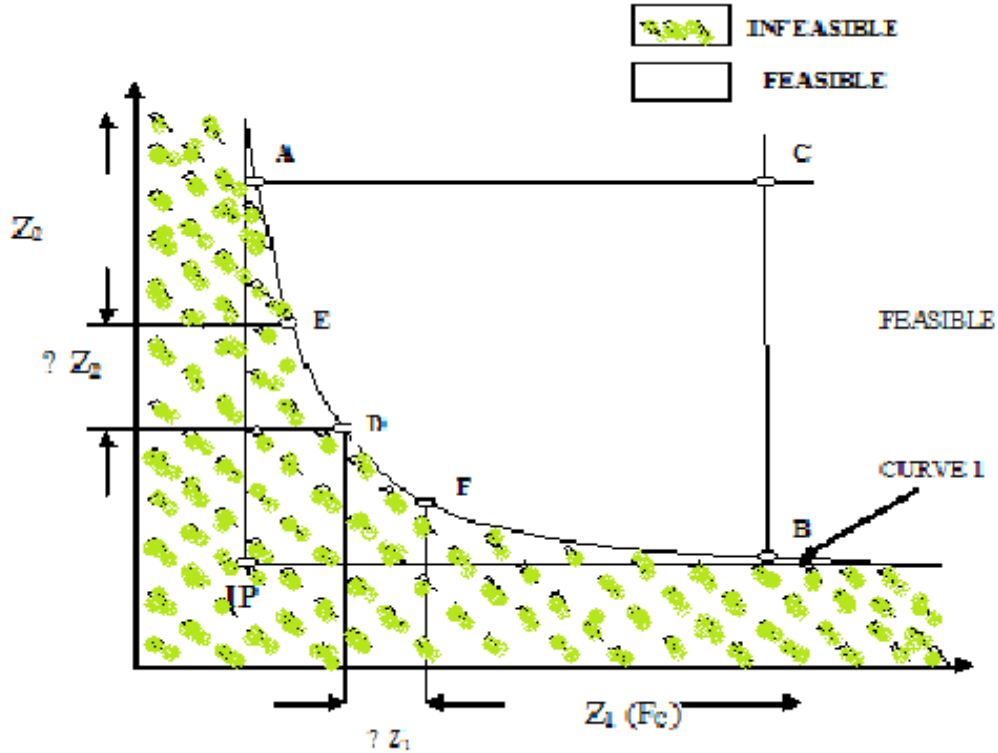


Fig 4.1: FEASIBLE REGION IN OBJECTIVE SPACE

#### 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**

### **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).

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

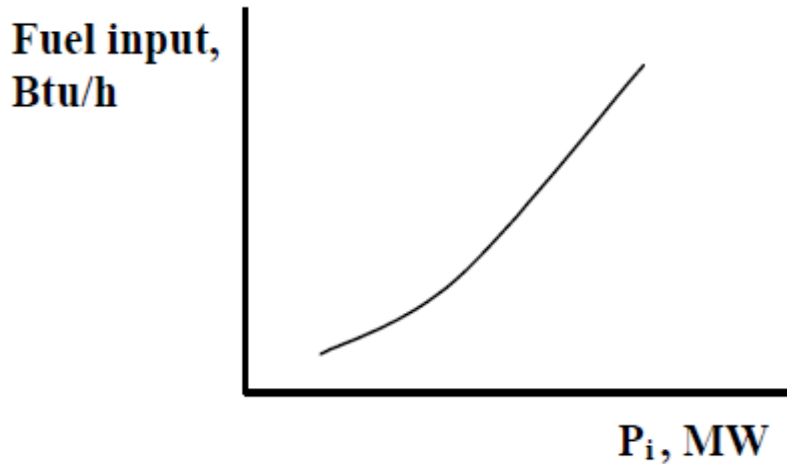
Where,  $i=1,2,\dots,N_g$

$N_g$  =number of generators

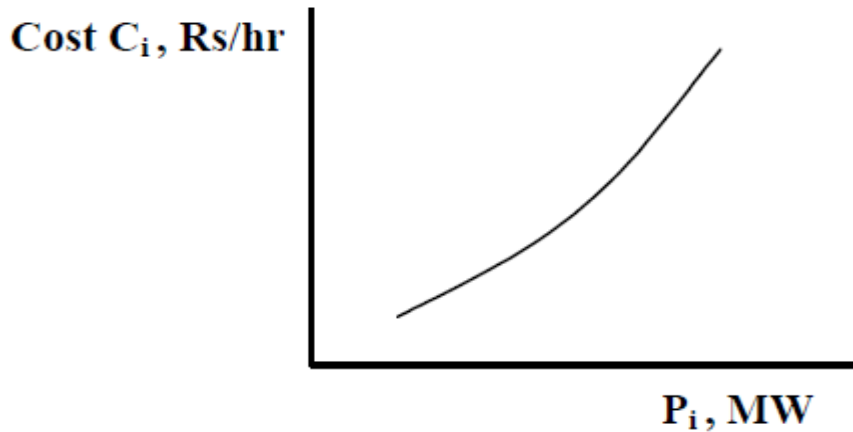
$a_i, b_i, c_i$  are the Cost coefficients of the  $i^{\text{th}}$  generating unit(see Appendix).

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 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.

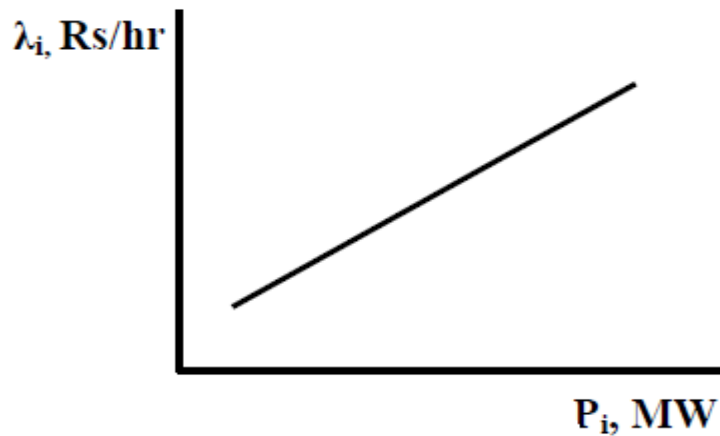


Fig. 5.1(c) Incremental fuel-cost curve

## 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, NxN 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  $i^{\text{th}}$  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

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

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  $i^{\text{th}}$  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  $i^{\text{th}}$  generator,  $C_i$  is the cost of generation for  $i^{\text{th}}$  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

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

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 14 bus system in 3D space.

TABLE 6.1

NONINFERIOR SET OF IEEE 14 BUS SYSTEM IN 3D SPACE

S.no	Cost of Generation		Transmission Loss		Environmental Pollution		Distance
	Wc	Fc	Wl	Fl	Wp	Fp	
1	1	<b>1132.43</b>	0	7.4761	0	641.4156	33.1602
2	1	1132.77	0	7.4223	0.1	637.2896	29.0373
3	1	1134.03	0	7.3065	0.2	628.6337	20.4463
4	1	1137.61	0	7.1466	0.5	617.4858	10.5999
<b>5</b>	<b>1</b>	<b>1141.33</b>	<b>0</b>	<b>7.0587</b>	<b>1</b>	<b>612.1371</b>	<b>9.7869</b>
6	1	1147.88	0	6.9772	5	608.5545	15.4714
7	1	1148.86	0	6.9629	8	608.3963	16.4476
8	1	1149.22	0	6.9552	10	608.3559	16.8066
9	1	1149.97	0	6.9394	20	608.2995	17.5551
10	1	1150.45	0	6.9297	50	608.2828	18.0343
11	1	1150.61	0	6.9264	100	608.2803	18.194
12	1	1150.71	0	6.9244	250	608.2797	18.2939
13	1	1150.75	0	6.9238	500	608.2796	18.3338
14	1	1132.43	0.5	7.4761	0	641.4156	33.1602



ECONOMIC LOAD DISPATCH FOR IEEE 14 BUS SYSTEM IN 3D SPACE

---

15	1	1132.84	0.5	7.4127	0.1	636.5634	28.3123
16	1	1134.12	0.5	7.3004	0.2	628.1879	20.0096
17	1	1137.68	0.5	7.1442	0.5	617.3332	10.5072
18	1	1141.38	0.5	7.0578	1	612.0882	9.7634
19	1	1147.89	0.5	6.9771	5	608.5525	15.4813
20	1	1148.88	0.5	6.9598	8	608.394	16.4674
21	1	1149.23	0.5	6.9527	10	608.3544	16.8165
22	1	1149.98	0.5	6.9381	20	608.2991	17.565
23	1	1150.45	0.5	6.9292	50	608.2827	18.0343
24	1	1150.62	0.5	6.9262	100	608.2803	18.204
25	1	1150.71	0.5	6.9243	250	608.2796	18.2939
26	1	1150.75	0.5	6.9237	500	608.2796	18.3338
27	1	1132.43	1	7.4761	0	641.4156	33.1602
28	1	1132.92	1	7.4035	0.1	635.8603	27.6109
29	1	1134.21	1	7.2944	0.2	627.7543	19.5859
30	1	1137.76	1	7.1419	0.5	617.1835	10.4189
31	1	1141.43	1	7.0569	1	612.0399	9.7905
32	1	1147.91	1	6.9757	5	608.5497	15.4912
33	1	1148.91	1	6.9567	8	608.3919	16.4872
34	1	1149.25	1	6.9502	10	608.353	16.8363
35	1	1149.99	1	6.9369	20	608.2987	17.575
36	1	1150.46	1	6.9287	50	608.2827	18.0442
37	1	1150.62	1	6.9259	100	608.2803	18.204
38	1	1150.72	1	6.9242	250	608.2796	18.3039
39	1	1150.75	1	6.9237	500	608.2796	18.3338
40	1	1150.05	5	6.937	20	608.297	17.6349
41	1	1150.48	5	6.9287	50	608.2824	18.0642
42	1	1150.63	5	6.9245	100	608.2802	18.2139
43	1	1150.72	5	6.9235	250	608.2796	18.3038
44	1	1150.75	5	6.9233	500	<b>608.2795</b>	18.3338

ECONOMIC LOAD DISPATCH FOR IEEE 14 BUS SYSTEM IN 3D SPACE

45	1	1150.11	8	6.9197	20	608.2969	17.6842
46	1	1150.51	8	6.9217	50	608.2824	18.0839
47	1	1150.64	8	6.9224	100	608.2802	18.2238
48	1	1150.72	8	6.9228	250	608.2796	18.3038
49	1	1150.75	8	6.923	500	608.2795	18.3338
50	1	1155.47	250	6.4682	10	612.2561	23.3821
51	1	1152.06	250	6.733	50	608.7014	21.3025
52	1	1151.43	250	6.8159	100	608.4042	19.6414
53	1	1151.04	250	6.8767	250	608.3018	19.01
54	1	1150.91	250	6.8993	500	608.2853	18.6219
55	1	1161.81	500	6.2141	0	631.5039	18.4928
56	1	1151.43	500	6.8159	0.1	608.4043	37.4429
57	1	1151.04	500	6.8767	0.2	608.3018	19.01
58	1	1150.91	500	6.8993	0.5	608.2853	18.6219
59	1	1159.18	500	6.2993	8	617.2771	35.0365
60	1	1158.54	500	6.3239	10	615.9726	28.2228
61	1	1156.23	500	6.4297	20	612.4109	27.22
62	1	1153.51	500	6.6093	50	609.5783	24.1569
63	1	1152.23	500	6.7299	100	608.7144	21.1237
64	1	1151.37	500	6.8338	250	608.3644	19.8115
65	1	1151.08	500	6.8761	500	608.3023	18.9504
66	1	1163.05	1000	<b>6.2122</b>	0	631.5712	18.6618
67	1	1153.51	1000	6.6093	0.1	609.5783	38.4719
68	1	1152.23	1000	6.73	0.2	608.7143	21.1237
69	1	1151.37	1000	6.8338	0.5	608.3644	19.8115
70	1	1162.95	1000	6.2134	1	629.4911	18.9504
71	1	1161.16	1000	6.2381	10	620.2809	37.1672
72	1	1152.02	1000	6.76	250	608.5809	31.1359
73	1	1151.41	1000	6.8332	500	608.3656	19.6

## 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 cost of generation ( $F_C$ ) with transmission loss ( $F_L$ ) 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 ( $F_P$ ) & transmission loss ( $F_L$ ) with environmental pollution ( $F_P$ ) respectively.

### (i) Cost of Generation vs Environmental Loss

Fig-6.1 represents the variation of cost of generation with the transmission loss. The X- axis represents the cost of generation and the Y- axis represents transmission loss.

The curve shows the behaviour of  $F_C$  &  $F_L$  in different ranges. From Fig 6.1, it is observed that when the cost of generation increases from 1130 \$/hr to 1162 \$/hr, the transmission loss decreases from 7.461 MW to 6.2 MW.

From Fig. 1 it is observed that minimum cost of generation  $F_{Cmin} = 1132$  \$/hr . At this value transmission loss is maximum and has a value of  $F_{Lmax} = 7.464$  MW. Similarly, when the cost of generation is maximum  $F_{Cmax} = 1162$  \$/hr, at this value the transmission loss is minimum and has a value of  $F_{Lmin} = 6.2$  MW.

Thus, it can be concluded that the cost of generation ( $F_C$ ) and transmission loss ( $F_L$ ) are ‘**conflicting**’ in nature. But for a small range, when the cost of generation is between 1149 to 1151 \$/hr, the behaviour between the two objectives is not very clear. It seems that the two objectives are still conflicting but to a lesser degree.

### (ii) Cost of Generation vs Environmental Pollution :

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

The curve shows the behaviour of ( $F_C$ ) & ( $F_P$ ) in different ranges. The curve shows that when cost of generation increases from 1132 \$/hr to 1150 \$/hr, the environmental pollution decreases from 641 kg/hr to 608 kg/hr which shows that they are ‘**conflicting**’ in nature in this range.

When the cost of generation further increases, the environmental pollution also increases. In other words when the cost of generation is in the range of 1150 \$/hr to 1162 \$/hr, the environmental pollution also increases from 608kg/hr to 622 kg/hr, which means they show the '**supportive**' behaviour in this range.

Thus, it can be concluded that the generation cost and environmental pollution may not always follow the conflicting behaviour. They can be conflicting in nature in one range and supportive in other range.

### (iii) **Transmission Loss vs Environmental pollution**

Fig-6.3 represents the variation of system transmission loss with the variation in environmental pollution. X- Axis represents the transmission loss of system. Y- Axis represents the environmental pollution.

The curve shows the behaviour  $F_L$  &  $F_P$  in different ranges. When the transmission loss increases from 6.2 MW to 6.9 MW, then the environmental pollution decreases from 631 kg/hr to 608 kg/hr, which means that they are '**conflicting**' in this range.

When the transmission loss further increases from 6.9MW to 704 MW, the environmental pollution increases 608 kg/hr to 641 kg/hr which means they are '**supportive**' in this range.

Thus, it can be concluded that the transmission loss and environmental pollution may not always follow the conflicting or supportive behaviour. They can be conflicting in nature for some range and supportive in other range.

### 6.3 3D GRAPHS:

Fig-6.4 & Fig 6.5 show the variation of the cost of generation ( $F_C$ ), Transmission Loss ( $F_L$ ), Environmental Pollution ( $F_P$ ), in 3D space. The curve APB shows the noninferior set in 3D space. The region above curve APB is feasible region and all solutions in this region are inferior. The region below the curve APB is infeasible region. The Ideal Point (1132.43 \$/hr, 6.212 mw/hr, 608.2795 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 AP and curve PB.

**CURVE AP** : It shows that all the three objectives  $F_C$ ,  $F_L$ ,  $F_P$  are conflicting in this region AP .

**CURVE PB**: The generation cost and transmission loss are relatively more conflicting as the slope of the curve is more in this region.

The generation cost and environmental pollution & environmental pollution and transmission loss are supportive in this region. In region  $P_1PP_2$  the objectives are less conflicting as compared to curve AP and PB.

It is, therefore concluded the 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 ( $F_{Cmin}= 1132.43$  \$/hr,  $F_{Lmin}=6.2122$  MW,  $F_{Pmin} = 608.2795$  kg/hr). It is observed that in case of 3D individual minimization of an objective function may not give its minimum value. Therefore, Target 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 5 in Table 6.1

2D and 3D graphs have been plotted from the computational results 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 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. 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 marked as TP and is seen closest to the origin (Ideal Point) for IEEE 14 bus system. 3D curve is also drawn in Fig 6.4

However it can be realized that the Ideal Point as defined and located above cannot be achieved in practice, which means that there cannot be any choice of weights which can give the minimum value of all objectives 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. The distance of all the noninferior points has been calculated and depicted in column 8 of Table 6.1 for IEEE 14 bus system. The Target point which is at the minimum distance from the Ideal Point is at the S.No. 5 giving the following values for the individual objective functions for 14 bus system:

$$F_C = 1141.33 \text{ \$/hr.}$$

$$F_L = 7.0567 \text{ MW.}$$

$$F_P = 612.1371 \text{ kg/hr.}$$

## **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 14 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**

Following are the suggestions for the future research work:

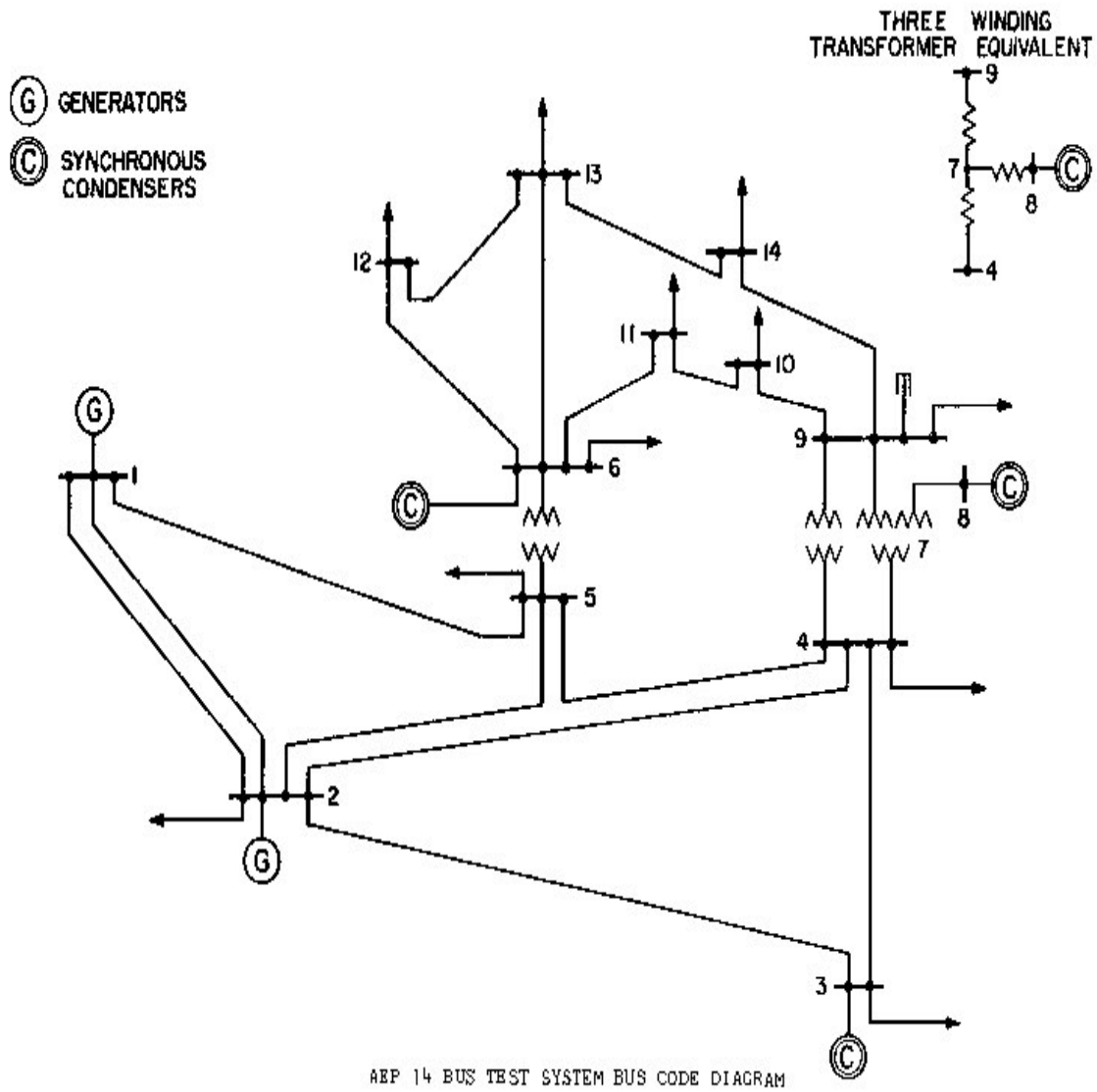
1. The objectives considered in the present research work are – Cost of Generation ( $F_C$ ), System Transmission Losses ( $F_L$ ) and Environmental Pollution ( $F_p$ ) which are noncommensurable because of their nature. However, transmission losses and environmental pollution can be expressed in monetary units to be compatible with the cost of generation. This objective function can be minimized to reach the Target Point (TP) or the best compromise solution.

2. Neural networks can be used to predict the load demand and to identify the noninferior set from a set of feasible solutions.
3. Interactive multiobjective programming techniques should be developed which can identify the Target Point (TP) in a single step.



**APPENDIX**

**IEEE 14 BUS SYSTEM:**



**Fig. Bus-Code Diagram 14 Bus System**

**TABLE I**

**Impedance and Line-Charging Data (14 Bus System)**

Line Designation	Resistance p.u.*	Reactance p.u.*	Line Charging	Tap Setting
1-2	0.01938	0.05917	0.0264	1
1-5	0.05403	0.22304	0.0246	1
2-3	0.04699	0.19797	0.0219	1
2-4	0.05811	0.17632	0.0187	1
2-5	0.05695	0.17388	0.0170	1
3-4	0.06701	0.17103	0.0173	1
4-5	0.01335	0.04211	0.0064	1
4-7	0	0.20912	0	1
4-9	0	0.55618	0	1
5-6	0	0.25202	0	1
6-11	0.09498	0.19890	0	1
6-12	0.12291	0.25581	0	1
6-13	0.06615	0.13027	0	1
7-8	0	0.17615	0	1
7-9	0	0.11001	0	1
9-10	0.03181	0.08450	0	1
9-14	0.12711	0.27038	0	1
10-11	0.08205	0.19207	0	1
12-13	0.22092	0.19988	0	1
13-14	0.17093	0.34802	0	1

\* Impedance and line-charging susceptance in p.u. on a 100 MVA base. Line charging one-half of total charging of line.

**TABLE II**

**BUS DATA or Operating Conditions (14 Bus System)**

Bus No.	Magnitude p.u.	Phase Angle deg.	Generation	Generation	Load	Load
			MW	MVAR	MW	MVAR
1*	1.06	0	0	0	0	0
2	1	0	40	0	21.7	12.7
3	1	0	0	0	94.2	19.0
4	1	0	0	0	47.8	-3.9
5	1	0	0	0	7.6	1.6
6	1	0	0	0	11.2	7.5
7	1	0	0	0	0	0
8	1	0	0	0	0	0
9	1	0	0	0	29.5	16.6
10	1	0	0	0	9.0	5.8
11	1	0	0	0	3.5	1.8
12	1	0	0	0	6.1	1.6
13	1	0	0	0	13.5	5.8
14	1	0	0	0	14.9	5.0

\* Slack Bus

**TABLE I-F**

**REGULATED BUS DATA(14 Bus System)**

Bus No.	Voltage Magnitude p.u.	Minimum MVAR capability	Maximum MVAR capability
2	1.045	-40	50
3	1.010	0	40
6	1.070	-6	24
8	1.090	-6	24

**Cost Characteristics:**

$$C_1 = 50 P_1^2 + 245 P_1 + 105 \text{ \$/hr}$$

$$C_2 = 50 P_2^2 + 351 P_2 + 44.4 \text{ \$/hr}$$

$$C_6 = 50 P_6^2 + 389 P_6 + 40.6 \text{ \$/hr}$$

Here the total load demand of the system is 259 MW. Maximum and minimum active power constraint on the generator bus for the given system is 150 MW and 50 MW respectively. Voltage magnitude constraint for generator bus 2 is 1.045, for bus no. 6 is 1.070, for bus no. 3 is 1.010 & for bus no. 8 is 1.090

**Emission Characteristics**

$$E_1 = 126 P_1^2 - 135.5 P_1 + 22.893 \text{ \$/hr}$$

$$E_2 = 137.5 P_2^2 - 124.9 P_2 + 137.37 \text{ \$/hr}$$

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

**B-Coefficients are:**

$$B_{11} = 0.0231$$

$$B_{12} = 0.0078$$

$$B_{13} = - 0.0007$$

$$B_{21} = 0.0078$$

$$B_{22} = 0.0182$$

$$B_{23} = 0.0022$$

$$B_{31} = - 0.0007$$

$$B_{32} = 0.0022$$

$$B_{33} = 0.0329$$

**M- File for 3D PROBLEM:**

**Objective Function M-file:**

Function z = objective14bus(x)

$$z = ((50*(x(1)/100)*(x(1)/100)) + (245*(x(1)/100)) + 105 + (50*(x(2)/100)*(x(2)/100)) + (351*(x(2)/100)) + 44.4 + (50*(x(3)/100)*(x(3)/100)) + (389*(x(3)/100)) + 40.6);$$

**Constraint Function M-file:**

Function [c,ceq]=constraint14bus(x)

$$c = [-x(1)+50; x(1)-150; -x(2)+50; x(2)-150; -x(3)+50; x(3)-150];$$

$$ceq = (x(1)+x(2)+x(3))*((0.000231*x(1)*x(1)) + (0.000156*x(1)*x(2)) + (0.00014*x(1)*x(3)) + (0.000044*x(2)*x(3)) + (0.000182*x(2)*x(2)) + (0.000329*x(3)*x(3))) - 259);$$

**M-file For Calculating B- Coefficients:**

clear

basemva=100;

accuracy=0.0001;

maxiter=10;

busdata=[1 1 1.06 0 0 150 0 0 0 0 0;2 2 1.045 0 21.7 12.7 63.11 0 -40 50 0;3 0 1.01 0 94.2 19 0 0  
0 40 0;4 0 1 0 47.8 -3.9 0 0 0 0 0;5 0 1 0 7.6 1.6 0 0 0 0 0;6 2 1.07 0 11.2 7.5 77.12 0 -6 24 0;7 0  
1 0 0 0 0 0 0 0 0;8 0 1.09 0 0 0 0 0 -6 24 0;9 0 1 0 29.5 16.6 0 0 0 0 0;10 0 1 0 9 5.8 0 0 0 0 0;11  
0 1 0 3.5 1.8 0 0 0 0 0;12 0 1 0 6.1 1.6 0 0 0 0 0;13 0 1 0 13.5 5.8 0 0 0 0 0;14 0 1 0 14.9 5 0 0 0  
0 0];

linedata=[1 2 0.01938 0.05917 0.0264 1;1 5 0.05403 0.22304 0.0246 1;2 3 0.04699 0.19797  
0.0219 1;2 4 0.05811 0.17632 0.0187 1;2 5 0.0595 0.17388 0.0170 1;3 4 0.06701 0.17103  
0.0173 1;4 5 0.01335 0.04211 0.0064 1;4 7 0 0.20912 0 1;4 9 0 0.55618 0 1;5 6 0 0.25202 0 1;6  
11 0.09498 0.19890 0 1;6 12 0.12291 0.25581 0 1;6 13 0.06615 0.13027 0 1;7 8 0 0.17615 0 1;7  
9 0 0.11001 0 1;9 10 0.03181 0.08450 0 1;9 14 0.12711 0.27038 0 1;10 11 0.08205 0.19207 0  
1;12 13 0.22092 0.19988 0 1;13 14 0.17093 0.34802 0 1];

disp(busdata)

disp(linedata)

mwlimits=[50 150;50 150;50 150];

lfybus

lfnewton

busout

bloss

## **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.