CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Economic load dispatch (ELD) is an important function in power system planning and operation. The basic object of economic load dispatch is the distribution of total generation of power in the network such that the cost of power delivered is minimum. By economic load dispatch we mean to find the generation of the different generators or plants so that the total fuel cost is minimum and at the same time the total demand and the losses at any instant must be met by the total generation. In case of economic load dispatch the generations are not fixed but they are allowed to take values again within certain limits so as to meet a particular load demand with minimum fuel consumption. This means economic load dispatch problem is really the solution of large number of load flow problems and choosing the one which is optimum in the sense that it needs minimum cost of generation.

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 30 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 30 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 etal [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 etal [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 etal [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 etal [46] convert the emission objective into a cost function and Kumaran and Mouly [10] convert the minimization of losses and the cost objective, along with the load balance constraint, into an index value and then all index values are summed into a single fitness function. The second technique is to use a specific multiobjective method. In this technique, all objective functions are in competition and a search algorithm is used to find an optimal solution. In noninferior set, a solution cannot be improved upon without adversely affecting the other objectives. Therefore, the result will be a set of optimal solutions that can be presented in a trade-off curve among all objectives. The second approach is used by Abido [4] and Yalcinoz [47].

1.4 PLAN OF THESIS

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

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

Chapter 2: Presents Genetic Algorithms and its applications.

Chapter 3: Minimization of Rosenbrock function manually using GA.

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 30 bus system.

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

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

Appendix and References are at the end of the thesis.

CHAPTER 2

GENETIC ALGORITHM

2.1 INTRODUCTION

In engineering disciplines a large spectrum of optimization problems have grown in size and complexity. In some instances, the solution for complex multidimensional problems by using classical optimization technique is difficult or expensive. This realization has lead 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 the based on the Darwinian thinking of natural selection and natural genetics. In general GAs start with an initial set of random solutions that lie in the feasible solution space. This random cluster of solution point is called the population. Each solution in the population represents a possible solution to the optimization problem therefore called the chromosome. The chromosome is a string of symbols based on the uniqueness of two state machines; they are commonly binary bit string.

2.2 DEFINITIONS AND CONCEPTS USED IN GENETIC COMPUTATION

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

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

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

Table 2.1 Terminology in Genetic Algorithms

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

2.2.1 Evolutionary Algorithms:

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

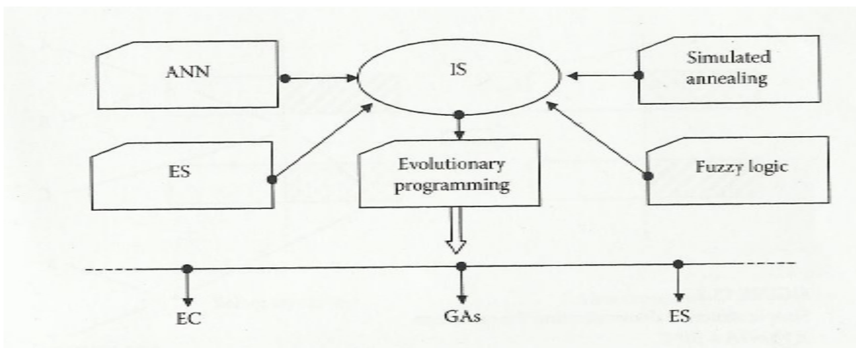

Fig 2.1 Common Classifications of IS

2.3 GA APPROACH

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

Stage 1: Creating an initial population.

Stage 2: Evaluating the fitness function.

Stage 3: Creating new populations.

2.3.1 GA Operators:

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

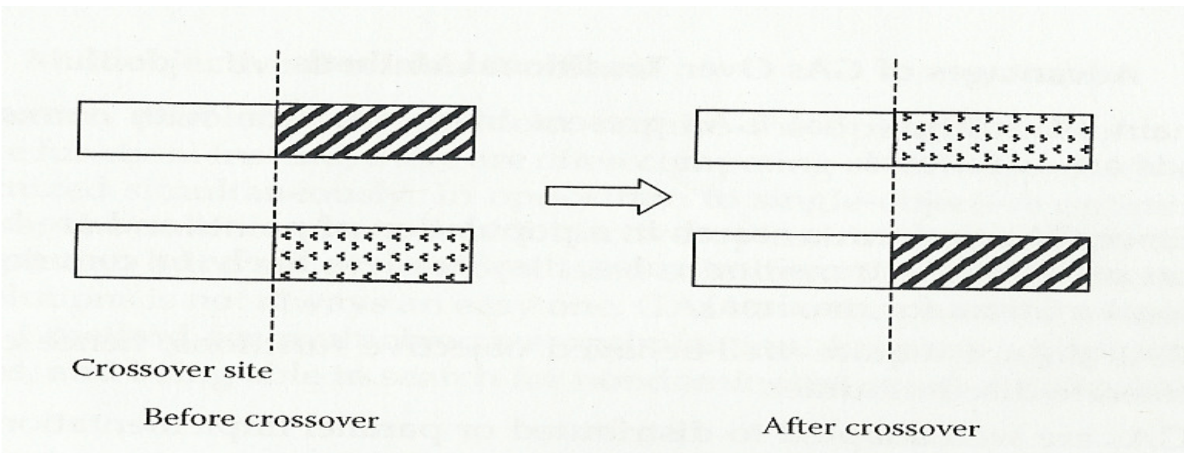

Fig 2.2 Crossover operation on a pair of strings

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

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

2.3.2 Major Advantages:

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

1. GAs do not have many mathematical requirements for optimization problems. Due to their evolutionary nature, GAs will search for solutions without regard to specific inner workings of the problem. They can handle any kind of objective functions, and any kind of constraints (i.e. linear or non-linear) defined on discrete, continuous, or mixed search spaces.

2. Ergodicity of evolution operators makes GAs very effective at performing global searches (in probability). The traditional approaches perform local search by a convergent stepwise procedure, which compares the values of nearby points and moves to the relative optimal points. Global optima can be found only if the problem possesses certain convexity properties that essentially guarantee that any local optima is a global optima.

3. GAs provide us with a great flexibility to hybridize with domain dependent heuristics to make an efficient implementation for a specific problem.

2.3.3 Advantages of GAs over Traditional Methods:

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

1. Since GAs perform a search in a population of points and are based on probabilistic transition rules, they are less likely to converge to local minima (or maxima).

2. GAs do not require well-behaved objective functions, hence easily tolerate discontinuities.

3. GAs are well adapted to distributed or parallel implementations.

4. GAs code parameters in a bit string and not in the values of parameters. The meaning of the bits is completely transparent for the GA.

5. GAs search from a population of points and not from a single point.

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

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

2.4 THEORY OF GAs

2.4.1 Constraints

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

2.4.2 Other GA Variants

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

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

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

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

2.4.3 Coding

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

2.4.4 Fitness

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

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

2.4.5 Selection

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

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

   We have used the Roulette wheel approach.

2.4.6 Crossover

The basic operator for producing new chromosome is crossover. In this operator, information is exchanged among strings of mating pool to create new strings. The aim of the crossover operator is to search the parameter space. Crossover is a recombination operator, which proceeds in three steps. First, the reproduction operator selects at random a pair of two individual string for mating, then a crossover site is selected at random along the string length and the position values are swapped between two strings following the cross site. There are many types of crossover as Single point crossover, two point crossover, Multipoint crossover, Uniform crossover, Matrix crossover etc. In the single point crossover, two individual strings are selected at random from the mating pool. Next, a crossover site is selected randomly along the string length and binary digits (alleles) are swapped between the two strings at crossover site. Suppose site 3 from left is selected at random. It means starting from the 4th bit and onwards, bits of strings will be swapped to produce offspring which is given by-

Single point crossover operation

Parent 1: X_1 = { 0 1 0 **1 1 0 1 0 1 1** }

Parent 2: X_2 = { 1 0 0 **0 0 1 1 1 0 0** }

Offspring 1: X_1 = { 0 1 0 **0 0 1 1 1 0 0** }

Offspring 2: X_2 = { 1 0 0 **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 **1 1 0** 1 0 1 1}

Parent 2: X_2 = {1 0 0 **0 0 1** 1 1 0 0}

Offspring 1: X_1 = {0 1 0 **0 0 1** 1 0 1 1}

Offspring 2: X_2 = {1 0 0 **1 1 0** 1 1 0 0}

2.4.7 Mutation

The final genetic operator in the algorithm is mutation. In general evolution, mutation is a random process where one allele of a gene is replaced by another to produce a new genetic structure. Mutation is an important operation, because newly created individuals have no new inheritance information and the number of alleles is constantly decreasing. This process results in the contraction of the population to one point, which is wished at the end of convergence process. Diversity is one goal of the learning algorithm to search always in regions not viewed before. Therefore, it is necessary to enlarge the information contained in the population. One way to achieve this goal is **mutation**. The role of mutation is often seen as providing a guarantee that the probability of searching any given string will never be zero and acting as safety net to recover good genetic material that may be lost through the action of selection and crossover. In GAs mutation is randomly applied with low probability in the range of 0.001 & 0.01 and modifies elements in the chromosome. Here, binary mutation flips the value of the bit at the loci selected to be the mutation point. Given that mutation is applied uniformly to an entire population of strings, it is possible that a given string may be mutated at more than one point.

Mutation operation

Offspring X_1: 1 1 **1** 1 0 1 0

New offspring X_2: 1 1 **0** 1 0 1 0

2.4.8 Parameters

Like other optimization methods, GAs have certain parameters such as

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

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

2.5 GENERAL ALGORITHM OF GAs

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

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

A new generation is formed by

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

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

CHAPTER 3

MINIMIZATION OF ROSENBROCK FUNCTION MANUALLY USING GA

3.1 INTRODUCTION TO ROSENBROCK FUNCTION

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

Range $x_1 \in (0, 2)$

$x_2 \in (0, 2)$

Since G.A maximizes the function, so using transformation, we get

Maximize $f(x) = \frac{1}{1+F(x)}$

Here, we are using binary strings, which can be obtained by randomly tossing a coin considering head as '1' and tail as '0'.

So, a variable 'x' whose bounds are given by x^L and x^U is represented by a string of 'q' binary bits, and its decimal equivalent is,

$$X = x^L + \frac{x^L - x^U}{2^q - 1} * \left\{\sum_{k=0}^{q-1}(2^k b_k)\right\}$$

Where,

x^L=lower bound

x^U=upper bound

q=no. of bits in the string

k=bit position from R.h.s

b=k^{th} bit value (may be 0 or 1)

if a continuous variable is to be represented with Δx accuracy, then number of binary bits(q)in a string is computed as,

$$2^q \geq \left\{\frac{x^L - x^U}{\Delta x}\right\} + 1$$

Let accuracy, Δx = 0.01(for this problem)

Then,

$2^q \geq \{\frac{(2-0)}{0.01}\} + 1 \geq 201$

This gives, q=8

Let the parent string x_1, as shown in column 2, s.no1 of table 1 is given as x1=01001101

Its decimal equivalent

$X = 0 + \frac{2-0}{2^8-1} * \left\{\sum_{k=0}^{7-1}(2^k b_k)\right\}$

$= \frac{2}{255} * \{\sum_{k=0}^{7}(2k * bk)\}$

$= \frac{2}{255}$ *[20*1+21*0+22*1+23*1+24*0+25*0+26*1+27*0]

$= \frac{2}{255}$ *[2+0+4+8+0+64]

$= \frac{2}{255}$*77

X=0.604

3.2 Solution performed manually using Genetic Algorithm:-

3.2.1 Iteration 1:-

Reproduction:- Rosenbrock function is a function of two variables x_1 and x_2. Reproduction is the first operation applied on population. In table 1, there are 8 columns, column 1 representing serial no. from 1 to 5. Column 2 and column 3 represents parent string x_1 and x_2 as population. Column 4 and 5 gives the information of decimal equivalent of x_1 and x_2 respectively. Column 6 gives the value of function f(x) obtained after substituting the decimal value of x_1 and x_2 in the given Rosenbrock function. Column 7 gives the ratio of respective value of f(x) to the average of f(x), and finally column 8 depicts the Actual count which means the probability of participation of parent string to move to the next iteration. As a thumb rule, since the total population set taken at the time of first iteration must not vary much, therefore the total number of strings taken to the next iteration should not be more or less than 1 from no. of strings taken for first iteration. Thus on seeing the column 7 the value which is very small then 1 is taken as Actual count '0' and as Actual count equal to "nearest whole number" for other values of $\bar{f}$.

Table 1-Reproduction (Result of iteration 1)

S.no 1	Population 2 x_1	 3 x_2	X_1 4	X_2 5	f(x) 6	$\bar{f}$ 7	Actual Count 8
1	01001101	00110110	0.604	0.424	0.666351	2.2275	2
2	00100010	01110001	0.267	0.384	0.088345	0.2951	1
3	10000100	10000001	1.035	1.012	0.739634	2.4719	2
4	00000111	11001100	0.055	1.600	0.003891	0.0130	0
5	11110001	00000111	1.890	0.055	0.000807	0.0027	0
					Sum=1.4961		
					Average=0.2992		

Since, actual count for 4^{th} and 5^{th} population is Zero. That's why these are discarded.

The avg. value of function f(x) comes out to be 0.2992 and it is shown in the last row of table 1.The difference between the max value of f(x) i.e 0.739634 as shown in column 6 s.no 3 and the min. value of f(x) ie 0.000807 as shown at column 5 s.no 5 comes out to be 0.738893, which is greater than required accuracy of 0.01

Thus the next iteration is carried out by performing crossover and mutation process on old population set of iteration 1and thus new population obtained acts as parent population for next iteration. This is shown in column 4 and 5 of table2.

Since the actual count for parent string 1 of table 1 is 2, it is selected twice for crossover. The process of crossover of string 1 is carried out with string 2 and 3 at 5^{th} place from the left. It is shown in table 2.

Crossover & mutation:-

Crossover process is done for parent string 1 with parent string 2at site 5^{th} place from left and for parent string 1 with parent string 3 at site 4^{th} place from left.

Mutation process has been carried out on string 3 in two ways. On x1 it is done by interchanging the bits at 6^{th} and 7^{th} position. On x2 it is done by flipping the bit at 8^{th} place.

Table 2(Crossover & mutation)

Old Population		Site	New Population(offspring)	
x1	x2		x1	x2
0100**1101**	0011**0110**	4	0100**0010**	0011**0001**
0010**0010**	0111**0001**		0010**1101**	0111**0110**
0100**1101**	0011**0110**	4	0100**0100**	0011**0001**
1000**0100**	1000**0001**		1000**1101**	1000**0110**
10000**10**0	1000000**1**		10000**01**0	1000000**0**

Offspring produced after 1^{st} iteration now acts as parent population for 2^{nd} iteration.

3.2.2 Iteration 2:-

Table 3-Reproduction (Result of iteration 2)

S.no 1	Population 2 x1	 3 x2	X1 4	X2 5	f(x) 6	$\bar{f}$ 7	Actual Count 8
1	01000010	00110001	0.5176	0.4235	0.273	0.727	1
2	00101101	01110110	0.3530	0.9255	**0.015**	0.034	0
3	01000100	00110001	0.5333	0.3843	0.451	1.201	1
4	10001101	10000110	1.1059	1.051	0.251	0.669	1
5	10000010	10000000	1.0196	1.004	**0.887**	2.363	2
				Sum= 1.877			
				Average=0.3754			

In this iteration, the average value of f(x) comes out to be 0.3754 which is greater than the avg. value of f(x) obtained in iteration 1 i.e 0.2992. Thus it means that we are moving in the right direction of maximizing the function f(x).

The difference of the max. value of f(x) i.e 0.887 and the min. value of f(x) i.e 0.015(both highlighted) is 0.872, which again is greater than required accuracy of 0.01. Thus the next iteration is performed. This process continues till the difference is smaller than the required accuracy.

Thus for next iteration new population set is obtained from old population set by performing crossover and mutation method. The new population set for 3^{rd} iteration is shown in table 4.

Table 4 (Crossover & mutation)

Old Population x1	x2	Site	New Population(offspring) x1	x2
010**00010**	001**10001**	5	010**00010**	001**00000**
100**00010**	100**00000**		100**00010**	100**10001**
0100**0100**	0011**0001**	4	0100**1101**	0011**0110**
1000**1101**	1000**0110**		1000**0100**	1000**0001**
100000**1**0	10000000		100000**0**0	10000000

3.2.3 Iteration 3:-

Table 5 -Reproduction (Result of iteration 3)

S.no 1	Population 2 x1	3 x2	X1 4	X2 5	f(x) 6	$\bar{f}$ 7	Actual Count 8
1	01000010	00100000	0.517	0.251	0.793	1.071	1
2	10000010	10010001	1.019	1.137	0.506	0.683	1
3	01001101	00110110	0.604	0.423	0.666	0.900	1
4	10000100	10000001	1.035	1.012	0.739	0.998	1
5	10000000	10000000	1.004	1.004	0.998	1.348	1
					Sum=3.7024		
					Average=0.740		

Table 6(Crossover & mutation)

Old Population x1	Old Population x2	Site	New Population(offspring) x1	New Population(offspring) x2
0**1000010**	0**0100000**	7	00000010	00010001
1**0000010**	1**0010001**		11000010	10100000
0100110**1**	0011011**0**	1	01001100	00110111
1000010**0**	1000000**1**		10000101	10000000
1000000**0**	1000000**0**		1000000**1**	1000000**1**

Again the average value of f(x) increases from 0.3754 to 0.740 thus, we are moving in right direction. Since, the difference between the max. and min. value of is greater than required accuracy therefore, we continue to next iteration with a set of new population obtained after performing crossover and mutation method.

3.2.4 Iteration 4:-

Table 7 -Reproduction (Result of iteration 4)

S.no	Population x1	Population x2	X1	X2	f(x)	$\bar{f}$	Actual Count
1	00000010	00010001	0.016	0.133	0.2681	0.5535	0
2	11000010	10100000	1.521	1.255	0.0088	0.0181	0
3	01001100	00110111	0.596	0.431	0.5755	1.1881	1
4	10000101	10000000	1.043	1.004	0.5853	1.2081	1
5	10000001	10000001	1.012	1.012	0.953	2.034	2
					Sum=2.422		
					Average=0.4844		

In this iteration, since the avg. value of f(x) decreases, the weak parent strings having actual count ‘0’ are discarded. So that, only the healthy strings are left for next iteration.

Table 8 (Crossover & mutation)

Old Population x1	Old Population x2	Site	New Population(offspring) x1	New Population(offspring) x2
01001100	00110111	alternate bit	00001001	00100011
10000001	10000001		11000100	10010101
10000101	10000000	alternate bit	10000101	10000001
10000001	10000001		10000001	10000000

Thus the new set of population is carried to next iteration by performing crossover along with mutation process on old population. The new population set thus obtained acts as population for the reproduction in next iteration. It is shown in table 8.

3.2.5 Iteration 5:-

Table 9 -Reproduction (Result of iteration 5)

S.no 1	Population 2 x1	Population 3 x2	X1 4	X2 5	f(x) 6	$\bar{f}$ 7	Actual Count 8
1	00001001	00100011	0.071	0.2745	0.120	0.270	0
2	11000100	10010101	1.537	1.168	0.007	0.015	0
3	10000101	10000001	1.035	1.004	0.688	1.549	2
4	10000001	10000000	1.012	1.004	0.960	2.162	3
					Sum=1.7758		
					Average=0.444		

Here, too the avg. f(x) further decreases from the previous value, giving the inference that even now the direction of the path is not the desired one. Hence, we move to the next iteration taking new set of population after performing crossover & mutation. The weak parent strings having actual count ‘0’ are discarded. So that, only the healthy strings are left for next iteration for better results. The new population set to be used for 6th iteration is shown in table 10.

Table 10 (Crossover & mutation)

Old Population x1	x2	Site	New Population(offspring) x1	x2
10**00**01**01**	10**00**00**00**	alternative pair	10**00**01**01**	10**00**00**00**
10**00**00**01**	10**00**00**00**		10**00**00**00**	10**00**00**00**
10**00**01**01**	10**00**00**00**	alternative pair	10**00**01**01**	10**00**00**00**
10**00**00**01**	10**00**00**00**		10**00**00**00**	10**00**00**00**
1000000**1**	1000000**0**		1000000**0**	1000000**1**

3.2.6 Iteration 6:-

Table 11 -Reproduction (Result of iteration 6)

S.no	Population x1	x2	X1	X2	f(x)	$\bar{f}$	Actual Count
1	10000101	10000000	1.043	1.004	0.586	0.703	1
2	10000000	10000000	1.004	1.004	0.998	1.998	1
3	10000101	10000000	1.043	1.004	0.586	0.703	1
4	10000000	10000000	1.004	1.004	0.998	1.998	1
5	10000000	10000001	1.004	1.011	0.999	1.999	1
					Sum=4.167		
					Average=0.833		

Since, avg. f(x) is increased from 0.444 obtained in 5th iteration to 0.833, it means we are moving in right direction. Thus the path is favourable. But since, the difference between the max. value of f(x) i.e 0.999 and the min. value of f(x) i.e 0.586 comes out to be 0.413. Which, is greater than the required accuracy of 0.01. Thus, next iteration is preformed.

Table 12(Crossover & mutation)

Old Population x1	Old Population x2	Site	New Population(offspring) x1	New Population(offspring) x2
1000**0000**	1000**0001**	4	1000**0000**	1000**0000**
1000**0000**	1000**0000**		1000**0000**	1000**0001**
10000**0**00	10000**0**00	5	10000**0**00	10000**0**00
10000**1**01	10000**0**00		10000**0**01	10000**1**00
10000**1**01	10000**0**00		10000**0**01	10000**1**00

3.2.7 Iteration 7:-

Table 13 -Reproduction (Result of iteration 7)

S.no 1	Population 2 x1	Population 3 x2	X1 4	X2 5	f(x) 6	$\bar{f}$ 7	Actual Count 8
1	10000000	10000000	1.0039	1.0039	0.9984	1.0049	1
2	10000000	10000001	1.0039	1.0117	0.9985	1.0050	1
3	10000000	10000000	1.0039	1.0039	0.9984	1.0049	1
4	10000001	10000100	1.0117	1.0353	0.9862	1.0027	1
				Sum=4.9677			
				Average=0.9935			

Here, the difference between the max. value of f(x) i.e 0.9985 and the min. value of f(x) i.e 0.9962 comes out to be 0.0023. Which is much less than the required accuracy i.e 0.01.

Thus, the iteration process is terminated and the avg. value of the function f(x) is taken as 0.9935

3.3 RESULTS

Iteration1:-

The green point shown in the figure is the optimum point i.e (1, 1) and rest of the points are obtained after 1st iteration. The graph clearly shows that the points obtained after 1st iteration are scattered around optimum point.

Iteration 2:-

The points obtained after iteration 2 are again scattered around optimum point (1,1).

Iteration 3:-

Scattering continues, thus no particular inference can be made regarding the most favourable value of the function.

Iteration 4:-

Hare the point ‘a’ which is too far from the desired optimum point is discarded, since it is the weakest parent string and thus directly affect the result by decreasing the average value of function f(x).

Iteration 5:-

Here some points appears to be very near to optimum point (1, 1), these points are the most healthy parent strings which would clearly increase the value of f(x) in next iteration. Also there are points (b, c) which are too far from optimum point(1,1). Hence, these points are discarded and are not carried to the next iteration.

Iteration 6:-

Here the points seem to have converged over optimum point but since the difference between the max. f(x) and min. f(x) value is greater than required accuracy (0.01) thus, next iteration is performed.

Iteration 7:-

All the points are converged at the optimum point (1, 1) within given accuracy limit of (0.01). Thus, the iteration process is terminated.

3.4 CONCLUSION

While solving Rosenbrock function using Genetic Algorithm manually, as we progress to higher iterations, an increase in average value indicates that we are proceeding in right direction and it is observed in first three iterations. After 3^{rd} iteration there is decrease in average value which indicates that we are not proceeding in right direction. Hence by discarding the weak parent strings, the avg. value of f(x) again increases in 5^{th} iteration.

There is decrease in the average values of f(x) due to the presence of weak parent string in the population set which adversely affect the result. Thus by discarding there strings and allowing only healthy strings to move to the next iteration the favourable path can be obtained.

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

$$Z(X_1, X_2, \ldots, X_n) = [Z_1(X_1, X_2, \ldots X_n); \tag{4.1a}$$

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

$$\ldots;$$

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

s.t.

$$g_i(X_1, X_2, \ldots X_n) \leq 0 \qquad i = 1,2,\ldots m \tag{4.1b}$$

$$X_j \geq 0 \qquad j = 1,2,\ldots n \tag{4.1c}$$

Where $Z(X_1,X_2,\ldots..X_n)$ is the multiobjective function and $Z_1(X_1,X_2\ldots\ldots X_n)$, $Z_2(X_1,X_2,\ldots\ldots X_n),\ldots\ldots\ldots Z_h(X_1,X_2,\ldots\ldots X_n)$ are the h individual objective functions. In the multiobjective function Z, the various individual functions $Z_1,Z_2,\ldots\ldots Z_h$ 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 $Z_1,Z_2,\ldots\ldots Z_h$ depending upon the approach.

Multiobjective approach to economic load dispatch has been carried out on IEEE 30 bus system in 3D space. The data of IEEE 30 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 Noninferiorty

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 ΔZ_2 (ΔF_L) units of Z_2 (F_L). Similarly, in moving from D to F, we have to sacrifice ΔZ_1 (ΔF_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

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

or

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

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

$$Z_K(Y) \leq Z_K(X) \quad \text{VK= 1, 2……H}$$

and

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

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.

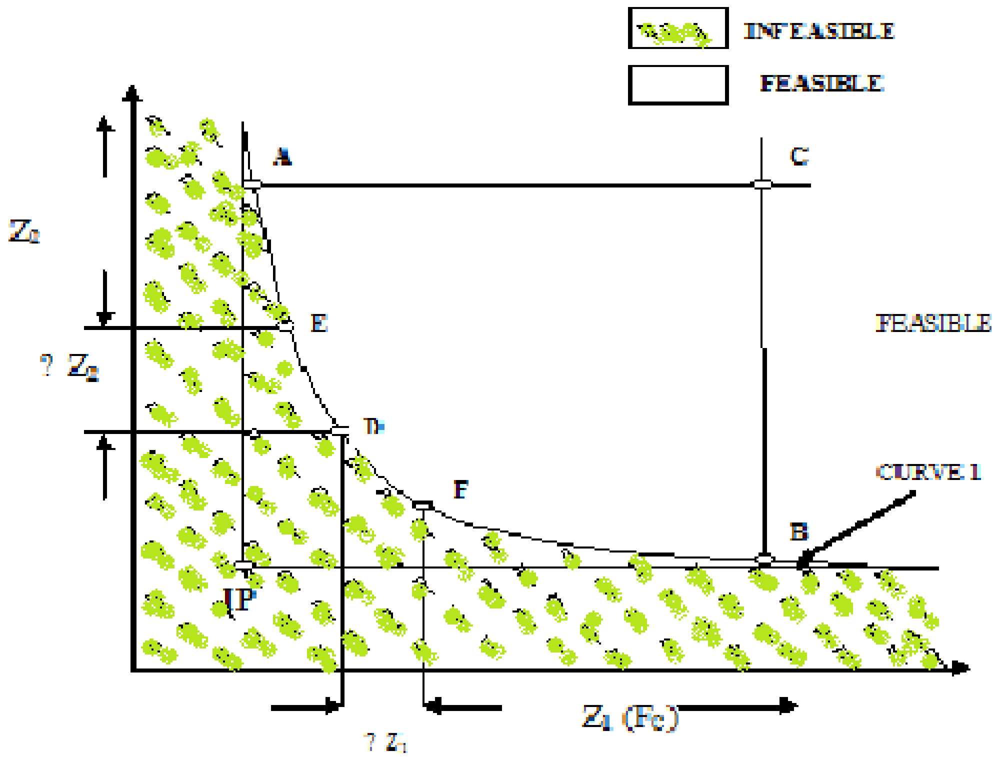

Fig 4.1: FEASIBLE REGION IN OBJECTIVE SPACE

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:

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

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).

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

Where, i=1,2,.......,N_g

N_g =number of generators

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

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

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

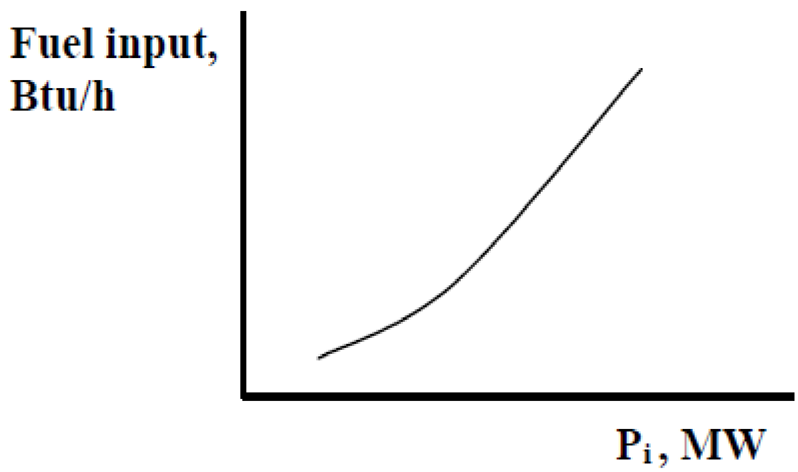

Fig. 5.1(a) Heat-rate curve

Fig 5.1(b) Fuel-rate curve

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

$$\frac{dC_i}{dP_i} = 2a_iP_i + b_i$$

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.

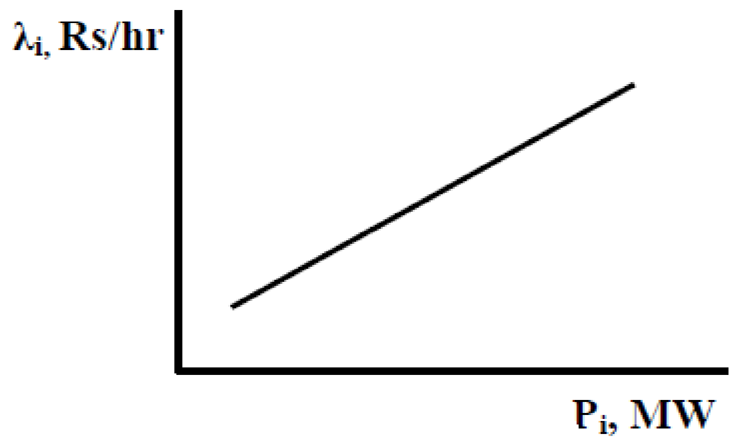

Fig. 5.1(c) Incremental fuel-cost curve

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

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

(i, j=1, 2,……,N_g)

Where i,j= number of generating units or plants i.e. i=j=1,2,3,….,Ng

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

$F_L = \Sigma\Sigma\ P_i\ B_{ij} P_j + \Sigma\ B_{oi} P_i + B_{oo}$

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:

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

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

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

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

P_D is power demand

F_L is function representing the system transmission loss.

5.5 ENVIRONMENTAL POLLUTION

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

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

Where, $i=1,2,.......,N_g$

N_g =number of generators and

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

5.6 FORMULATION OF MOELD PROBLEM IN 3D SPACE

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

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

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

$$F_C = \Sigma F[C_i(P_{gi})] \qquad (i=1,2,\ldots\ldots,N_g) \qquad (5.4)$$

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

The objective function to minimize system transmission loss is given as

$$F_L=\Sigma\Sigma\, P_i\, B_{ij}\, P_j \qquad (5.5)$$

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

N_g = number of generators.

The objective function to minimize environmental pollution is given as

$$F_p=\sum_{i=1}^{NG}(\, d_i P_i^2 + e_i\, P_i + f_i\,) \qquad (5.6)$$

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

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

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

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 30 bus system in 3D space.

TABLE 6.1

NONINFERIOR SET OF IEEE 30-BUS SYSTEM IN 3D SPACE

	Generation Cost($/hr)		Transmission Loss(MW)		Environmental Pollution(kg/hr)		
S.No	**Wc**	**Fc**	**Wl**	**Fl**	**Wp**	**Fp**	**DISTANCE**
1	1	1257.09	0	11.8024	0	675.4632	21.8418
2	1	1257.16	0	11.9078	0.1	673.8858	20.3322
3	1	1257.98	0	11.735	0.2	669.3574	15.9503
4	1	1260.63	0	11.243	0.5	661.0833	8.8748
5	**1**	**1263.42**	**0**	**10.8867**	**1**	**657.0847**	**8.0026**
6	1	1268.4	0	10.3998	5	654.3688	11.8245
7	1	1269.13	0	10.2276	8	654.2202	12.4768
8	1	1269.13	0	10.2276	10	654.2202	12.4768
9	1	1270.28	0	10.1837	50	654.1684	13.5793
10	1	1257.2	0.5	11.7664	0	676.0834	22.4394
11	1	1257.26	0.5	11.8575	0.1	674.0814	20.5105
12	1	1268.48	0.5	10.3843	1	654.3534	11.8964
13	1	1269.19	0.5	10.2117	5	654.2438	12.5307
14	1	1270.29	0.5	10.1821	50	654.1682	13.5886
15	1	1270.49	0.5	10.1766	500	654.166	13.7817
16	1	**1257.53**	1	11.6966	0.1	672.5811	19.0207
17	1	1258.56	1	11.4588	0.2	667.1421	13.8137
18	1	1268.56	1	10.369	5	654.3395	11.9684
19	1	1268.56	1	10.369	8	654.3395	11.9684
20	1	1269.98	1	10.2133	20	654.1783	13.2953

21	1	1270.3	1	10.1804	50	654.168	13.5979
22	1	1269.23	5	10.19	5	654.2772	12.564
23	1	1269.23	5	10.19	8	654.2772	12.564
24	1	1269.84	5	10.2122	10	654.1963	13.1594
25	1	1270.17	5	10.141	20	654.1739	13.4623
26	1	1270.17	5	10.141	50	654.1739	13.4623
27	1	1270.5	5	10.2251	500	654.166	13.8028
28	1	1267.91	8	10.5742	1	654.3504	11.4106
29	1	1270.01	8	10.1804	8	654.186	13.3164
30	1	1270.5	8	10.2034	250	654.1661	13.7977
31	1	1270.51	8	10.1721	500	**654.166**	13.8001
32	1	1268.68	10	10.329	0.5	654.3867	12.073
33	1	1270.24	10	10.1828	8	654.1688	13.5402
34	1	1270.51	10	10.1484	500	654.166	13.7946
35	1	1276.01	20	9.465	1	656.2508	19.1992
36	1	1271.24	20	10.0174	10	654.2281	14.4776
37	1	1270.53	20	10.176	500	654.1661	13.8204
38	1	1321.54	50	7.4915	0	700.3855	79.3116
39	1	1316.94	50	7.5884	0.1	695.4034	72.6839
40	1	1316.94	50	7.5884	0.2	695.4034	72.6839
41	1	1304.47	50	7.9113	0.5	682.2378	55.08
42	1	1295.64	50	8.2176	1	672.295	42.6187
43	1	1277.91	50	9.2585	5	657.5866	21.2244
44	1	1275.2	50	9.5108	8	655.8366	18.3655
45	1	1274.26	50	9.6408	10	655.3248	17.4173
46	1	1272.37	50	9.8991	20	654.43	15.5632
47	1	1271.24	50	10.0249	50	654.2275	14.4792
48	1	1356.87	100	6.9826	0	741.1399	132.365
49	1	1350.96	100	7.0453	0.1	733.7344	123.0557
50	1	1345.85	100	7.1057	0.2	727.4626	115.1119
51	1	1333.87	100	7.274	0.5	713.2193	96.8636
52	1	1320.55	100	7.5173	1	698.116	77.1951
53	1	1287.97	100	8.6052	5	665.1041	32.8015
54	1	1281.83	100	8.9851	8	660.0215	25.5044
55	1	1279.61	100	9.1534	10	658.3832	23.016
56	1	1275	100	9.5877	20	655.5349	18.1541
57	1	1272.25	100	9.9377	50	654.4272	15.4527
58	1	1271.37	100	10.015	100	654.2355	14.6042
59	1	1270.85	100	10.1633	250	654.1776	14.1289
60	1	1272.1	100	9.9198	500	654.4015	15.3017
61	1	1359.23	250	6.96	0.5	744.2642	136.1994
62	1	1358.77	250	6.964	1	743.5254	135.3659
63	1	1313.07	250	7.6976	5	689.4818	66.193
64	1	1300.34	250	8.0817	8	676.2956	48.5958
65	1	1295.2	250	8.2811	10	671.2941	41.8031
66	1	1283.33	250	8.9036	20	660.9101	27.1628

67	1	1275.47	250	9.5348	50	655.6829	18.622
68	1	1272.91	250	9.8576	100	654.5976	16.0898
69	1	1271.45	250	10.0169	250	654.241	14.6829
70	1	1270.98	250	10.1453	500	654.1853	14.2515
71	1	1359.58	500	6.9578	1	744.8297	136.8361
72	1	1339.1	500	7.1539	5	720.4141	105.4253
73	1	1323.54	500	7.463	8	700.8923	81.2355
74	1	1316.05	500	7.6269	10	692.5717	70.3685
75	1	1296.51	500	8.2322	20	672.3817	43.444
76	1	1281.12	500	9.0646	50	659.1819	24.6383
77	1	1275.63	500	9.5437	100	655.7359	18.7855
78	1	1272.47	500	9.9311	250	654.4578	15.6679
79	1	1265.14	500	9.8921	500	654.4564	8.5738
80	1	1360.55	1000	**6.9549**	0	746.4367	138.6285
81	1	1360.13	1000	6.9556	1	745.7413	137.8524
82	1	1358.96	1000	6.9622	5	743.8204	135.7034
83	1	1343.44	1000	7.1383	8	724.2403	111.206
84	1	1317.63	1000	7.5911	10	694.241	72.6051
85	1	1292.2	1000	8.4221	20	668.3139	37.8817
86	1	1292.2	1000	8.4221	50	668.3142	37.8818
87	1	1281.32	1000	9.0532	100	659.2932	24.8552
88	1	1274.62	1000	9.6561	250	655.238	17.7692
89	1	1272.5	1000	9.9191	500	654.4619	15.6952
90	1	1257.19	0	11.8024	0	675.4632	21.842
91	1	1257.23	1	11.7321	0	676.7166	23.0513
92	1	1263.42	0	10.8867	1	657.0847	8.0026

.

6.2 2D GRAPHS:

The results, by taking various combinations of three objective functions in Table 6.1 are represented by 2D graphs. The Cost of generation (F_c) with Transmission loss (F_L) graph is shown in Fig 6.1. Similarly, Fig 6.2 and Fig 6.3 shows 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 (F_C) vs Transmission loss (F_L):

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 1255 $/hr to 1365 $/hr, the transmission loss decreases from 11.9078 MW to 6.9556 MW.

From Fig. 1 it is observed that minimum cost of generation F_{Cmin} = 1257.16 $/hr . At this value transmission loss is maximum and has a value of F_{Lmax} =11.9078 MW. Similarly, when the cost of generation is maximum F_{Cmax} = 1360.55 $/hr, at this value the transmission loss is minimum and has a value of F_{Lmin} = 6.9556 MW.

Thus, it can be concluded that the cost of generation (F_c) and transmission loss (F_L) are '**conflicting**' in nature. Thus to decrease the transmission losses in the plant the cost of generation has to be increased.

(ii) Cost of Generation (F_C) v/s Environment pollution (F_P):

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 1255 $/hr to 1272 $/hr, the environmental pollution decreases from 678 kg/hr to 654.16 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 1272 $/hr to 1360 $/hr, the environmental pollution also increases from 654 kg/hr to 746.436 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 environment 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 6.9549 MW to 10.1721 MW, then the environmental pollution decreases from 746.4367 kg/hr to 654.166 kg/hr, which means that they are '**conflicting**' in this range.

When the transmission loss further increases from 10.1721 MW to 11.6966 MW, the environmental pollution increases 654.166 kg/hr to 672.58 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 and 6.5 show the variation of Cost of generation (F_C), Transmission Loss (F_L) and Environmental Pollution (F_P). The curve $A_4P_4B_4$ and $A_5P_5B_5$ show the noninferior set in 3D space. The region above the curves is feasible region and all solutions within this region are inferior. The region below the curves is infeasible region. The Ideal Point (1257.09$/hr, 6.9549 MW, 654.1660 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 curve A_5P_5 shows that all objectives F_L, F_C, F_P are conflicting in nature. Curve P_5B_5 shows that cost of generation (F_C) and environmental pollution (F_P) are supportive in nature. On the other hand transmission loss (F_L) and environmental pollution (F_P) are conflicting in nature. It is therefore concluded that the three objectives may not always follow the traditional behaviour, their behaviour can be different in different ranges.

6.4 DISCUSSION

The Ideal Point i.e.(F_{Cmin}, F_{Lmin}, F_{Pmin}) is located from the Table 6.1 as (F_{Cmin}= 1257.53 \$/hr, F_{Lmin}=6.9549 MW, F_{Pmin} = 654.1660 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 (Fcmin, Flmin, Fpmin). The Target Point is market as TP and is seen closest to the origin (Ideal Point) for IEEE 30 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 30 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 30 bus system:

F_C =1263.42 \$/hr.

F_L =10.8867 MW.

F_P =657.0847 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 30 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

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

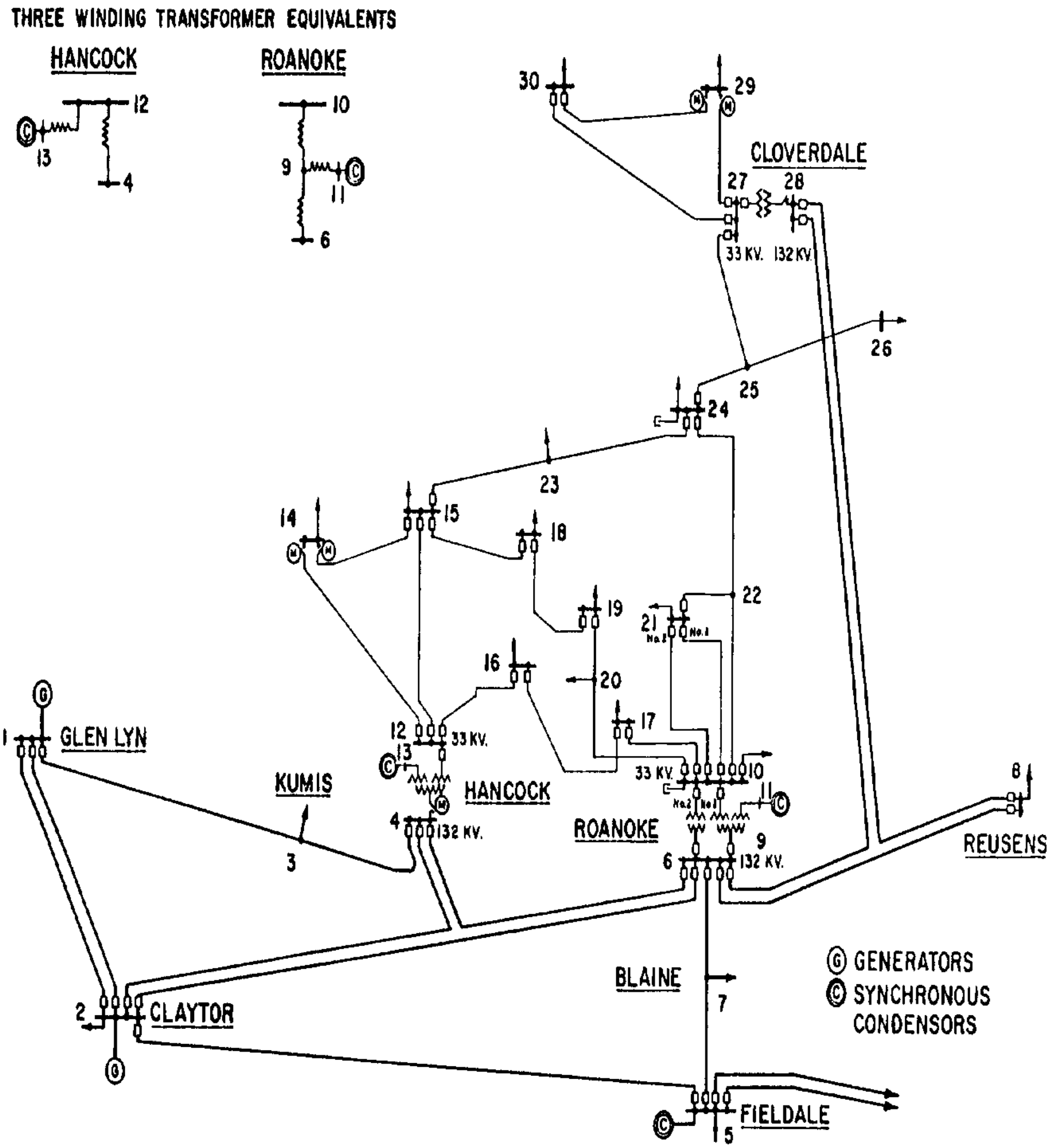

Fig. Bus-Code Diagram of IEEE 30 Bus System

TABLE I

IMPEDANCE or LINE-CHARGING DATA (30 Bus System)

Line Designation	Resistance p.u.*	Reactance p.u.*	Line Charging	Tap Setting
1-2	0.0192	0.0575	0.0264	1
1-3	0.0452	0.1852	0.0204	1
2-4	0.0570	0.1737	0.0184	1
3-4	0.0132	0.0379	0.0042	1
2-5	0.0472	0.1983	0.0209	1
2-6	0.0581	0.1763	0.0187	1
4-6	0.0119	0.0414	0.0045	1
5-7	0.0460	0.1160	0.0102	1
6-7	0.0267	0.0820	0.0085	1
6-8	0.0120	0.0420	0.0045	1
6-9	0	0.2080	0	0.978
6-10	0	0.5560	0	0.969
9-11	0	0.2080	0	1
9-10	0	0.1100	0	1
4-12	0	0.2560	0	0.932
12-13	0	0.1400	0	1
12-14	0.1231	0.2559	0	1
12-15	0.0662	0.1304	0	1
12-16	0.0945	0.1987	0	1
14-15	0.2210	0.1997	0	1
16-17	0.0824	0.1923	0	1
15-18	0.1070	0.2185	0	1

18-19	0.0639	0.1292	0	1
19-20	0.0340	0.0680	0	1
10-20	0.0936	0.2090	0	1
10-17	0.0324	0.0845	0	1
10-21	0.0348	0.0749	0	1
10-22	0.0727	0.1499	0	1
21-22	0.0116	0.0236	0	1
15-23	0.1000	0.2020	0	1
22-24	0.1150	0.1790	0	1
23-24	0.1320	0.2700	0	1
24-25	0.1885	0.3292	0	1
25-26	0.2544	0.3800	0	1
25-27	0.1093	0.2087	0	1
27-28	0	0.3960	0	0.968
27-29	0.2198	0.4153	0	1
27-30	0.3202	0.6027	0	1
29-30	0.2399	0.4533	0	1
8-28	0.0636	0.2000	0.0214	1
6-28	0.0169	0.0599	0.0065	1

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

TABLE II

BUS DATA or Operating Conditions (30 Bus System)

Bus No.	Magnitude p.u.	Phase Angle Degrees	Generation MW	Generation MVAR	Load MW	Load MVAR
1*	1.06	0	0	0	0	0
2	1	0	40	0	21.7	12.7
3	1	0	0	0	2.4	1.2
4	1	0	0	0	7.6	1.6
5	1	0	0	0	94.2	19.0
6	1	0	0	0	0	0
7	1	0	0	0	22.8	10.9
8	1	0	0	0	30.0	30.0
9	1	0	0	0	0	0
10	1	0	0	0	5.8	2.0
11	1	0	0	0	0	0
12	1	0	0	0	11.2	7.5
13	1	0	0	0	0	0
14	1	0	0	0	6.2	1.6
15	1	0	0	0	8.2	2.5
16	1	0	0	0	3.5	1.8
17	1	0	0	0	9.0	5.8
18	1	0	0	0	3.2	0.9
19	1	0	0	0	9.5	3.4
20	1	0	0	0	2.2	0.7
21	1	0	0	0	17.5	11.2
22	1	0	0	0	0	0
23	1	0	0	0	3.2	1.6

24	1	0	0	0	8.7	6.7
25	1	0	0	0	0	0
26	1	0	0	0	3.5	2.3
27	1	0	0	0	0	0
28	1	0	0	0	0	0
29	1	0	0	0	2.4	0.9
30	1	0	0	0	10.6	1.9

* Slack Bus

TABLE III

<u>Regulated Bus Data (30 Bus System)</u>

Bus Number	Voltage Magnitude p.u.	Minimum MVAR Capability	Maximum MVAR Capability
2	1.045	-40	50
5	1.01	-40	40
8	1.01	-10	40
11	1.082	-6	24
13	1.071	-6	24

TABLE IV

Transformer Data (30 Bus System)

Transformer Designation	Tap Setting*
4-12	0.932
6-9	0.978
6-10	0.969
28-27	0.968

* Off-nominal turns ratio, as determined by the actual transformer-tap positions and the voltage bases. In the case of nominal turns ratio, this would equal 1.

TABLE V

Static Capacitor Data (30 Bus System)

Bus Number	Susceptance* p.u.
10	0.19
24	0.043

* Susceptance in p.u. on 100 MVA base.

Cost Characteristics:

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

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

$C_8 = 50 P_8\text{^}2 + 389 P_8 + 40.6$ $/hr

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. 5 is 1.01, for bus no. 8 is 1.010, for bus no. 11 is 1.082 & for bus no. 13 is 1.071

Emission Characteristics

$E_1 = 135.5 P_1\text{^}2 – 126.5 P_1 +22.9$ \$/hr

$E_2 = 124.8 P_2\text{^}2 - 137.8 P_2 + 137.3$ \$/hr

$E_8 = 80.5 P_8\text{^}2 – 76.7 P_8 + 367.7$ \$/hr

The noninferior set for the above system has been obtained by solving the following problem.

Minimize $F_{(e)}$

st.

$50<=P_i<=150$ for i = 1, 2, 8

$$P_D + P_L - \sum_{n}^{NG} P_n = 0$$

$F_L \leq L_1$

B-Coefficients Calculated are as:

B11 = 0.0231

B12 = 0.0078

B13 = - 0.0007

B21 = 0.0078

B22 = 0.0182

B23 = 0.0022

B31 = - 0.0007

B32 = 0.0022

B33 = 0.0329

M- File For 3-D PROBLEM:

Objective Function M-file:

Function z = objective30busel(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]=constraint30busel(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))283.4(100*(((x(1)/100)*(x(1)/100)*0.0307)+(2*(x(1)/100)*(x(2)/100)*0.0129)+(2*(x(1)/100)*(x(3)/100)*(0.0002))+((x(2)/100)*(x(2)/100)*0.0152)+(2*(x(2)/100)*(x(3)/100)*(0.0011))+((x(3)/100)*(x(3)/100)*0.0190)));((135.5*(x(1)/100)*(x(1)/100))+(126.5*(x(1)/100))+22.9+(124.8*(x(2)/100)*(x(2)/100))+(137.8*(x(2)/100))+137.3+(80.5*(x(3)/100)*(x(3)/100))+(-76.7*(x(3)/100))+363.7)(specified-emission)];
(100*(((x(1)/100)*(x(1)/100)*0.0307)+(2*(x(1)/100)*(x(2)/100)*0.0129)+(2*(x(1)/100)*(x(3)/100)*(0.0002))+((x(2)/100)*(x(2)/100)*0.0152)+(2*(x(2)/100)*(x(3)/100)*(-0.0011))+
((x(3)/100)*(x(3)/100)*0.0190-specified loss)));

<u>M-file For Calculating B-Coefficients</u>:

clear

basemva=100;

accuracy=0.0001;

maxiter=10;

busdata=[1 1 1.06 0 0 0 0 0 0 0 0; 2 2 1.045 0 21.7 12.7 90 0 -40 50 0; 3 0 1 0 2.4 1.2 0 0 0 0 0; 4 0 1 0 7.6 1.6 0 0 0 0 0; 5 0 1.01 0 94.2 19 0 0 -40 40 0; 6 0 1 0 0 0 0 0 0 0 0; 7 0 1 0 22.8 10.9 0 0 0 0 0; 8 2 1.01 0 30 30 150 0 -10 40 0; 9 0 1 0 0 0 0 0 0 0 0; 10 0 1 0 5.8 2 0 0 0 0 0.19; 11 0

1.082 0 0 0 0 0 -6 24 0; 12 0 1 0 11.2 7.5 0 0 0 0 0; 13 0 1.071 0 0 0 0 0 -6 24 0; 14 0 1 0 6.2 1.6 0 0 0 0 0; 15 0 1 0 8.2 2.5 0 0 0 0 0; 16 0 1 0 3.5 1.8 0 0 0 0 0; 17 0 1 0 9 5.8 0 0 0 0 0; 18 0 1 0 3.2 0.9 0 0 0 0 0; 19 0 1 0 9.5 3.4 0 0 0 0 0; 20 0 1 0 2.2 0.7 0 0 0 0 0; 21 0 1 0 17.5 11.2 0 0 0 0 0; 22 0 1 0 0 0 0 0 0 0 0; 23 0 1 0 3.2 1.6 0 0 0 0 0; 24 0 1 0 8.7 6.7 0 0 0 0 0.043; 25 0 1 0 0 0 0 0 0 0 0; 26 0 1 0 3.5 2.3 0 0 0 0 0; 27 0 1 0 0 0 0 0 0 0 0; 28 0 1 0 0 0 0 0 0 0 0; 29 0 1 0 2.4 0.9 0 0 0 0 0; 30 0 1 0 10.6 1.9 0 0 0 0 0];

linedata=[1 2 0.0192 0.0575 0.0264 1; 1 3 0.0452 0.1852 0.0204 1; 2 4 0.0570 0.1737 0.0184 1; 3 4 0.0132 0.0379 0.0042 1; 2 5 0.0472 0.1983 0.0209 1; 2 6 0.0581 0.1763 0.0187 1; 4 6 0.0119 0.0414 0.0045 1; 5 7 0.0460 0.1160 0.0102 1; 6 7 0.0267 0.0820 0.0085 1; 6 8 0.0120 0.0420 0.0045 1; 6 9 0 0.2080 0 0.978; 6 10 0 0.5560 0 0.969; 9 11 0 0.2080 0 1; 9 10 0 0.1100 0 1 ; 4 12 0 0.2560 0 0.932; 12 13 0 0.1400 0 1; 12 14 0.1231 0.2559 0 1; 12 15 0.0662 0.1304 0 1; 12 16 0.0945 0.1987 0 1; 14 15 0.2210 0.1997 0 1; 16 17 0.0824 0.1923 0 1; 15 18 0.1070 0.2185 0 1; 18 19 0.0639 0.1292 0 1; 19 20 0.0340 0.0680 0 1; 10 20 0.0936 0.2090 0 1; 10 17 0.0324 0.0845 0 1; 10 21 0.0348 0.0749 0 1; 10 22 0.0727 0.1499 0 1; 21 22 0.0116 0.0236 0 1; 15 23 0.1000 0.2020 0 1; 22 24 0.1150 0.1790 0 1; 23 24 0.1320 0.2700 0 1; 24 25 0.1885 0.3292 0 1; 25 26 0.2544 0.3800 0 1; 25 27 0.1093 0.2087 0 1; 27 28 0 0.3960 0 0.968;27 29 0.2198 0.4153 0 1; 27 30 0.3202 0.6027 0 1; 29 30 0.2399 0.4533 0 1;8 28 0.0636 0.2000 0.0214 1; 6 28 0.0169 0.0599 0.0065 1];

disp(busdata)

disp(linedata)

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, N0. 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.